

CRANFIELD UNIVERSITY

Albert Stevan Johan van Heerden

**Computational Techniques for
Aircraft Evolvability Exploration
during Conceptual Design**

SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING

PhD

Academic Year: 2018/19

Supervisors:

Prof. Marin D. Guenov

and

Dr Arturo Molina-Cristóbal

4 February 2019

CRANFIELD UNIVERSITY
SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING

PhD

Academic Year: 2018/19

Albert Stevan Johan van Heerden

Computational Techniques for Aircraft
Evolvability Exploration during
Conceptual Design

Supervisors:

Prof. Marin D. Guenov

and

Dr Arturo Molina-Cristóbal

4 February 2019

This thesis is submitted in partial fulfilment of the requirements for
the degree of Doctor of Philosophy

© Cranfield University, 2019. All rights reserved. No part of this
publication may be reproduced without the written permission of
the copyright owner.

“Die Schlange, welche sich nicht häuten kann, geht zu Grunde.”

“The snake that cannot shed its skin perishes.”

Friedrich Nietzsche, *Morgenröte*, §573.

Abstract

Evolvability is a critical consideration during the design of an aircraft. It refers to the extent to which a baseline design could be reused, or ‘easily’ modified to create descendant designs that would meet future requirements. Since a major fraction of the cost of an aircraft programme is determined by decisions made during conceptual design, it is essential that the design space is explored thoroughly during this stage to find evolvable designs.

Existing computational methods to perform such exploration exist, but are limited in two respects. The first of these is that, with existing techniques, derivatives are usually generated by applying pre-specified modifications to a selected baseline, such that each derivative in a study is only linked to a single baseline. The designer must therefore evaluate large numbers of baseline-derivative pairs to adequately capture the evolution options available. The second limitation concerns an absence of appropriate down-selection criteria to narrow down the number of design points when evolvability is considered.

The work presented in this thesis addresses these limitations. The aim was to develop computational techniques that would enable aircraft designers to explore the evolvability of their designs more efficiently and effectively during the conceptual design stage. The scope was limited to civil transport aircraft and specifically to airframes. The work is applicable to both single- and twin-aisle aircraft, but the focus was, to a small degree, more on single-aisles. The research resulted in two main contributions: 1) a framework to provide a means to link all derivatives to all the baselines; and 2) a set of techniques to filter out inferior designs systematically.

The framework builds on the premise that the degree of ‘similarity’ between two designs could be used as an estimate for the redesign effort (i.e. resource expenditure) required to change one of these into the other. Case studies involving existing aircraft families were conducted to determine which design changes could be considered ‘reasonable’. Based on this information, a set of techniques to assess airframe similarity was developed, which involves automatically predicting possible commonality across two designs. Several algorithms were devised to achieve this, including one that solves a longest common subsequence problem to find common body segments and a simple optimisation procedure to find common wing elements. Notably, these techniques can be used to compare aircraft with dissimilar

configurations. For testing purposes, the framework was applied to several existing and future aircraft. The results showed that the predicted commonality matches published information regarding commonality and design re-use between designs. The framework essentially removes the need to model each future design option based on a specific starting design.

The design filtering techniques involve the application of set-based design to facilitate systematic down-selection of potential designs. Specifically, it is demonstrated how established set-based design criteria could be adapted to prune an evolvability design space progressively.

To demonstrate the usefulness of the research, it was applied to an example, concerning design candidates for a new single-aisle, environmentally friendly passenger aircraft. The results of this study were presented to a panel of design specialists from Airbus UK. The panel concluded that the proposed similarity assessment provides reasonable initial estimates for redesign effort and that the overall approach adds value to the evolvability exploration process.

Keywords: Aircraft Conceptual Design; Evolvability; Changeability; Set-Based Design; Knowledge-Based Engineering; Multi-Attribute Tradespace Exploration; Longest Common Subsequence.

Acknowledgements

What a journey this has been... Sitting back now, and reflecting on the past few years, I am astonished by how many people have helped me throughout my PhD study (as well as helping me prepare for it). Nearing the end of this journey, I need to express my sincere gratitude to the following people:

Pappa en Mamma. Waar moet ek begin? Ek is só dankbaar dat ek julle tweetjies as ouers gekry het. As ek enige sê gehad het in die proses van ouers kies, het ek darem regtig waar goed gekies. Baie, baie dankie dat julle my groot gemaak het soos julle het. Dirk en Ruthea. Duisend dankies vir julle liefde en ondersteuning. Alhoewel ons wêrelde van mekaar is, het dit vir my altyd gevoel of julle net hier by my is om my te ondersteun. Aan Ouma Lena, ek is baie dankbaar dat Ouma my altyd met soveel liefde moed in praat — dit help my altyd om dinge met nuwe energie aan te pak. Dankie ook aan die res van my familie — ek is baie lief vir julle almal.

My supervisors, Professor Marin Guenov and Dr Arturo Molina-Cristóbal. I am eternally grateful for your leadership, guidance, occasional (and necessary) nudging, and friendship. I have learned so much from you and I do hope I can repay you by being an excellent researcher myself. My colleagues in the AEDG, past and present, I thank you for the great atmosphere you have helped to create in the office. Our discussions, movie nights, and nation-themed dinners made my time as a PhD student so much more fun than it would have been otherwise. Thank you, Atif Riaz, Yogesh Bile, Xin Chen, Gabriele Mura, Sergio Jimeno Altelarrea, Soufianne El-Fassi, Mario Carta, Vitaly Voloshin, and Varun Datta. I would also like to thank my friends from other research groups, particularly Jiravud Klongtrujrok and Abudurehman Bilale.

I could not have completed this work without the help of so many other people. Indeed, a special thank you to Teresa Townsend (I am grateful for your support and pragmatism), Emma Turner, Brian Riddle from the National Aerospace Library, Dr Dominique Fleischmann, Diane Boyles, Elizabeth James, and Dr Craig Lawson. To all the amazing experts at Airbus with whom I had the privilege to work, including Chris Lynas, Dr Sanjiv Sharma, and Adrian Murton — thank you for your time, advice, and invaluable feedback regarding my work. In addition, I need

to thank the following people from Airbus for their support, enthusiasm for my work, and friendship: Ian Marr, Ranjit Ravindranath, and Dr Etienne Coetzee.

My friends from Caltech, especially Neel Nadkarni, Chris Roh, Trent Kirchdoerfer, David Huynh, Nathan Martin, and Stephanie and Sarah Mitchell. Thank you so much for making my time in California such a memorable experience. Dr Felicia Hunt, I am indebted to you. Neel, you have kept me sane and I am grateful for your friendship!

Werner, dankie vir jou vriendskap. Ek sien altyd só baie uit na ons geselsies. En nou begin jy met jou studie!

My fellow researchers around the world. It has been a privilege learning from you.

All my teachers, past and present. Thank you so much for passing on the knowledge of our wonderful world and encouraging me to think critically. I would particularly like to express my gratitude to professors Beverley McKeon, Josua Meyer, Leon Liebenberg, and Tunde Bello-Ochende.

Shaun Schoeman. Jy het vir my 'n deur oopgemaak wat my wêreld verander het. Jy leef voort in my werk.

Contents

Abstract	v
Acknowledgements	vii
List of Figures	xv
List of Tables	xix
List of Algorithms	xxi
Abbreviations	xxiii
Symbols	xxv
1 Introduction	1
1.1 Betting the Company – Aircraft Evolvability and its Importance . .	1
1.2 Motivation	4
1.2.1 Computational Expense	4
1.2.2 Lack of Systematic Filtering Procedures	11
1.3 Aim and Objectives	11
1.4 Scope	13
1.5 Research Methodology	14
1.5.1 Research Clarification	15
1.5.2 Descriptive Study I	15
1.5.3 Prescriptive Study	16
1.5.4 Descriptive Study II	16
1.6 Thesis Structure	17
2 Literature Review	19
2.1 Introduction	20

2.2	Aircraft Design	20
2.2.1	Overview of the Aircraft Design Process	20
2.2.2	The Conceptual Design Phase	21
2.3	Computational Approaches	22
2.3.1	Design Space Exploration	22
2.3.2	Geometry Modelling and Parameterisation	24
2.3.3	Knowledge-Based Engineering	25
2.4	Engineering Systems Evolvability	26
2.4.1	Engineering Change	27
2.4.2	Engineering Systems Changeability	28
2.4.3	The Concept of Evolvability	30
2.5	Evolution as a Phenomenon in Aircraft	31
2.5.1	Beginnings	32
2.5.2	Military Examples	32
2.5.3	Evolution in Civil Jet Transport Aircraft	34
2.5.4	Unconventional Configurations	35
2.5.5	‘Variant’, ‘Version’, ‘Derivative’, and ‘New’ Aircraft	36
2.6	Design for Evolvability	38
2.6.1	Baseline Design	38
2.6.2	Uncertainty Recognition	39
2.6.3	Concept Generation	40
2.6.4	Evolvability Exploration	47
2.6.5	Process Management	55
2.6.6	Critical Discussion: Design for Evolvability	55
2.7	Similarity Assessment Techniques	59
2.8	Set-Based Design	61
2.8.1	Interval Dominance	62
2.8.2	Maximality	63
2.8.3	E-Admissibility	63
2.8.4	Discussion	64
2.9	Conclusions	64
3	Case Studies in Civil Transport Aircraft Evolvability	67
3.1	Introduction	68
3.2	Boeing	69
3.2.1	The progenitor – the Boeing 367-80	69
3.2.2	The Boeing 707/720	70
3.2.3	The Boeing 720	73
3.2.4	The Boeing 727	74
3.2.5	The Boeing 737 Family	75
3.2.6	The Boeing 747	80
3.2.7	The Boeing 757 and 767 Family	85
3.2.8	The 777 Family	87
3.2.9	The 787 Family	90

3.3	McDonnell Douglas	91
3.3.1	The DC-8	91
3.3.2	The DC-9 and descendant families	93
3.3.3	The DC-10 and MD-11	97
3.4	Airbus	98
3.4.1	The A300	98
3.4.2	The A310	100
3.4.3	The A320 Family	101
3.4.4	The A330 and A340 families	103
3.4.5	The A350 Family	107
3.4.6	The A380	107
3.5	Payload-range capability evolution strategies	108
3.5.1	Boeing	109
3.5.2	McDonnell Douglas	112
3.5.3	Airbus	113
3.5.4	Final Notes on Payload-Range Evolution	114
3.6	Modifications by Design Goal	115
3.6.1	Goal Type 1	117
3.6.2	Goal Type 2	119
3.6.3	Goal Type 3	120
3.6.4	Goal Type 4	121
3.6.5	Goal Type 5	123
3.7	Maximum Changes	123
3.8	Conclusions	123
4	Automatic Airframe Similarity Assessment	127
4.1	Introduction	128
4.2	Overview	128
4.3	Knowledge Representation	130
4.3.1	Aircraft Object Model	131
4.3.2	Aircraft Design Structure Matrix	139
4.4	Airframe Partitioning	144
4.5	Airframe Commonality Prediction	146
4.5.1	Airframe Body Components	148
4.5.2	Wings	157
4.5.3	Empennages	167
4.5.4	Undercarriage	168
4.5.5	Support Components	169
4.5.6	Commonality Aggregation	170
4.6	Conclusions	170
5	Set-Based Design Techniques for Evolvability Exploration	173
5.1	Introduction	174
5.2	Input Description	175

5.3	Elimination Criteria	177
5.3.1	Interval Dominance	178
5.3.2	Maximality Criterion	182
5.3.3	E-Admissibility Criterion	184
5.4	Constrain and Union Strategy	186
5.5	Conclusions	189
6	Evaluation	191
6.1	Introduction	192
6.2	Demonstration Case Study	193
6.2.1	Purpose and Overview	193
6.2.2	Airframe Configurations and Novel Technologies	194
6.2.3	Scenario Descriptions and Utility Functions	195
6.2.4	Modelling and Simulation	198
6.2.5	Results and Discussion	204
6.3	Airframe Similarity Assessment Framework Evaluation	208
6.3.1	Purpose	208
6.3.2	Software Implementation	209
6.3.3	Application to Existing Aircraft	210
6.3.4	Application to Dissimilar Configurations	217
6.3.5	Application to Demonstration Case Study	219
6.3.6	Computational Cost	232
6.4	Set-Based Techniques Demonstration	232
6.4.1	Purpose	232
6.4.2	Approach	232
6.4.3	Results and Discussion	233
6.5	Industry Feedback	237
6.5.1	Purpose	237
6.5.2	Approach	237
6.5.3	Results and Discussion	239
6.6	Conclusions	241
7	Summary and Conclusions	243
7.1	Introduction	243
7.2	Research Summary	243
7.2.1	Objective 1	245
7.2.2	Objective 2	248
7.2.3	Evaluation	249
7.3	Research Findings	251
7.4	Research Contributions	254
7.5	Future Work	256
	References	276

A	Civil Transport Aircraft Evolvability Data	277
B	MATLAB[®] Script for Example in the Text	289
	B.1 Fuselage Commonality Example	289
C	Evaluation: Software Implementation	293
	C.1 Case Study Model Descriptions	293
	C.1.1 Aircraft Sizing: FLOPS	293
	C.1.2 Configuration-Technology Combination Selector	293
	C.1.3 Control Surface Mass Calculation	293
	C.1.4 High-Lift Device Aerodynamics Prediction	294
	C.1.5 Cost	294
	C.1.6 Noise Calculator	294
	C.1.7 Utility Calculator	294
	C.2 Workflow Calibration	294
	C.3 Commonality Prediction Implementation	296
	C.3.1 Aircraft Instantiation	296
	C.3.2 Airframe Partitioning	297
	C.3.3 Body Commonality Prediction	297
	C.3.4 Wing Commonality Prediction	298
	C.3.5 Empennage Commonality Prediction	299
	C.3.6 Undercarriage Commonality Prediction	299
	C.3.7 Support Component Commonality Prediction	300
	C.3.8 Commonality Aggregation	300
	C.3.9 Commonality Plots	300
D	Evaluation Questionnaire	301
E	Publications by the Author	307

List of Figures

1.1	The Boeing 737 – a design that has undergone much evolution . . .	3
1.2	Evolvability evaluation	5
1.3	Evolvability assessment computation time	7
1.4	Novel aircraft configurations	10
1.5	Research methodology	15
1.6	Structure of the thesis.	17
2.1	The phases of aircraft design	21
2.2	Example of an aircraft sizing workflow	24
2.3	Geometries created using the CST representation.	25
2.4	Evolution of the F/A-18	33
2.5	BWB family commonality	35
2.6	Box-wing aircraft family.	36
2.7	Technological inheritance	37
2.8	Taxonomy of design for flexibility procedures	38
2.9	Taxonomy of evolution paths.	41
2.10	Reserves in undercarriage length.	44
2.11	The commonality spectrum	46
2.12	Overview of the EvoLVE method	49
2.13	Multi-attribute tradespace definition	51
2.14	System performance and expectations over time	52
2.15	Evolvability from the near- to far-future	54
2.16	Decision possibilities between two alternatives.	63
2.17	Variation of expected utility with uncertain parameter.	64
3.1	The Boeing model 367-80	69
3.2	Boeing 367-80 fuselage cross-section evolution	70
3.3	Boeing 707 wing evolution	73
3.4	Boeing 737-300 changes from the 737-200	77
3.5	Boeing 737 nacelle evolution	78
3.6	Changes across Boeing 747 variants	81

3.7	Differences between the 747-400 and 747-8 wings	84
3.8	Boeing 777-300/-300ER undercarriage at rotation	89
3.9	The 777X compared with the 777-300ER	89
3.10	DC-9 and MD-80 wing comparison	95
3.11	Airbus A300/A310 comparison	100
3.12	A321 wing showing chord extension	103
3.13	A330/A340 wing evolution	104
3.14	Boeing small-to-medium aircraft payload-range evolution.	110
3.15	Boeing medium-to-large aircraft payload-range evolution.	111
3.16	McDonnell Douglas aircraft payload-range evolution.	113
3.17	Airbus aircraft payload-range evolution.	114
3.18	Design changes for design goal type 1.	118
3.19	Design changes for design goal type 2.	119
3.20	Design changes for design goal type 3.	121
3.21	Design changes for design goal type 4.	122
3.22	Design changes for design goal type 5.	122
4.1	Overview of the automatic similarity assessment framework.	129
4.2	UML class diagram for the aircraft object model.	131
4.3	<i>AircraftComponent</i> and <i>AirframeComponent</i> class descriptions . . .	132
4.4	UML description of the <i>Body</i> object model.	134
4.5	<i>Body</i> class geometry rendering	135
4.6	UML description of the <i>LiftingSurface</i> model.	136
4.7	<i>LiftingSurface</i> class geometry	137
4.8	Attributes of the <i>LsSegment</i> class	138
4.9	Attributes of the lifting-surface station class	139
4.10	Generic aircraft design-structure matrix (DSM) format.	141
4.11	Overview of the commonality prediction module.	147
4.12	Example input for fuselage commonality prediction	154
4.13	Recursive array for the example.	155
4.14	Finding the longest common subsequence for the example.	155
4.15	Commonality prediction results for the example.	157
4.16	Wing planform comparison.	159
5.1	SBD elimination criteria.	177
5.2	Interval dominance	179
5.3	Interval dominance: configurations or attribute sets.	181
5.4	Maximality	182
5.5	E-admissibility	184
5.6	Set where no overall dominance is present.	187
5.7	Selecting ‘Fuzzy’ Pareto points.	187
5.8	Specifying minimum utility and maximum redesign effort.	188
5.9	Utility and redesign effort with two baselines	188
6.1	Configuration-technology combinations selected case study.	195

6.2	Demonstration case study computational workflow	200
6.3	MAT results of the sampling experiments (Part I)	206
6.4	MAT results of the sampling experiments (Part II)	207
6.5	Automatic commonality prediction workflow	209
6.6	Commonality prediction for the A320.	212
6.7	Commonality prediction for the A320 and A321.	213
6.8	Commonality prediction for the A340 and A330.	214
6.9	Commonality prediction: A340-300 and A340-600.	215
6.10	Commonality prediction for the 707 and 727.	215
6.11	Commonality prediction for the 727 and 737.	216
6.12	Commonality prediction: 737-200 and -400.	216
6.13	Commonality prediction: conventional and BLI.	218
6.14	Commonality prediction: conventional and Airbus future concept.	219
6.15	Commonality prediction: conventional and BWB.	219
6.16	Baseline-descendant commonality evolvability exploration matrix	221
6.17	Investigating descendant cost (conventional – Part I)	223
6.18	Investigating descendant cost (conventional – Part II)	224
6.19	Commonality results for a conventional baseline	225
6.20	Investigating descendant cost (t-tail – Part I).	226
6.21	Investigating descendant cost (t-tail – Part II).	227
6.22	Commonality results for a T-tail baseline	228
6.23	Exploring commonality for a selected baseline	229
6.24	Future to past exploration (single descendant)	230
6.25	Future to past exploration (multiple descendants)	231
6.26	Applying maximality to far-future designs	234
6.27	Applying E-admissibility to far-future designs	235
6.28	Applying the constrain and union strategy	236
6.29	Responses to Question 1	240

List of Tables

1.1	Mass and performance effects of commonality.	8
3.1	Major variants of the Boeing 707 family	71
3.2	Major variants of the Boeing 727 family.	75
3.3	Major variants of the Boeing 737 Original family.	76
3.4	Major variants of the Boeing 737 Classic family.	79
3.5	Major variants of the Boeing 737NG family.	80
3.6	Major variants of the Boeing 737 MAX family.	81
3.7	Major variants of the Boeing 747 family.	82
3.8	Major variants of the Boeing 757 family.	86
3.9	Major variants of the Boeing 767 family.	86
3.10	Major variants of the Boeing 777 family.	88
3.11	Major variants of the Boeing 787 family.	90
3.12	Major variants of the Douglas DC-8 family.	92
3.13	Major variants of the McDonnell Douglas DC-9 family.	94
3.14	Major variants of the McDonnell Douglas MD-80 family.	96
3.15	Major variants of the McDonnell Douglas DC-10 family.	97
3.16	Major variants of the Airbus A300.	99
3.17	Major variants of the Airbus A310.	101
3.18	Major variants of the Airbus A320 family.	102
3.19	Major variants of the Airbus A330 family.	105
3.20	Major variants of the Airbus A340 family.	106
3.21	Major variants of the Airbus A350 family.	107
3.22	Development times for different types of derivative design goals. . .	116
3.23	Maximum parameter value changes for aircraft surveyed.	124
4.1	Description of salient <i>AircraftComponent</i> attributes.	132
4.2	Description of salient <i>AirframeComponent</i> attributes.	133
4.3	Description of salient <i>Body</i> attributes.	135
4.4	Description of salient <i>LsSegment</i> class attributes.	137
4.5	Description of salient <i>LsStation</i> class attributes.	138

4.6	Aircraft component categories.	141
4.7	Description of Airframe DSM entry fields.	143
4.8	Subscripts for the constituent wing substructures.	167
6.1	NASA N+3 subsonic fixed-wing project metrics	194
6.2	Scenario utility weighting factors and constraints	197
6.3	Main computational models employed	200
6.4	Overall computational workflow input	201
6.5	Overall computational workflow output	201
6.6	Computational workflow intermediate variables	201
6.7	DoE set-up for combinations C1 and C3	203
6.8	DoE set-up for combinations C2a to C5d	204
6.9	DoE set-up for combinations C6a and C6b	204
6.10	Existing baseline-descendant aircraft pairs	211
6.11	Respondent information	238
A.1	Boeing aircraft surveyed.	278
A.2	McDonnell Douglas aircraft surveyed.	279
A.3	Airbus aircraft surveyed.	280
A.4	References consulted.	281
A.5	Change descriptions.	282
A.6	G1 Changes: wing and fuselage	283
A.7	G1 Changes: other components	284
A.8	G2 Changes: wing and fuselage	285
A.9	G2 Changes: other components	286
A.10	G3 Changes: wing and fuselage	286
A.11	G3 Changes: other components	287
A.12	G4 Changes: wing and fuselage	287
A.13	G4 Changes: other components	288
C.1	Workflow calibration results	295
C.2	Wing extension optimisation parameters	298
C.3	Allowable changes in tail surface areas	299

List of Algorithms

4.1	Creating a comparison matrix for components of category τ	145
4.2	Finding the body-LCS recursive array.	152
4.3	Finding x -locations corresponding to LCS	153
5.1	Interval dominance applied to baseline options.	180
5.2	Interval dominance: configurations or attribute sets	181
5.3	Maximality applied to baseline options.	183
5.4	E-admissibility applied to far-future options.	185

Abbreviations

AEDG	Advanced Engineering Design Group
BBT	Bright Bold Tomorrow (Scenario)
CAEP	Committee on Aviation Environmental Protection
CROR	Counter-rotating open-rotor
CS	Certification Specification
CST	Class-function/Shape-function Transformation
CTC	Configuration-Technology-Combination
DRM	Design Research Methodology
DoE	Design-of-Experiments
DS	Design Space
DS	Descriptive Study
DSM	Design Structure Matrix
EASA	European Aviation Safety Agency
EIS	Entry-Into-Service
EvoLVE	Evaluation of Lifelong Vehicle Evolution
F3STF	Future three-Shaft TurboFan
FAA	Federal Aviation Administration
FAR	Federal Aviation Regulation
FLOPS	Flight Optimisation System
GW	Gross Weight
ICAO	International Civil Aviation Organization
KBE	Knowledge-Based Engineering
KC	King Carbon (Scenario)

LCS	Longest Common Subsequence
MAT	Multi-Attribute Tradespace
MATE	Multi-Attribute Tradespace Exploration
MLW	Maximum Landing Weight
MTOW	Maximum Take-Off Weight
NASA	National Aeronautics and Space Administration
NeoCASS	Next generation Conceptual Aero-Structural Sizing Suite
NiMBY	Not in My Back Yard (Scenario)
OEM	Original Equipment Manufacturer
OEW	Operating Empty Weight
PS	Prescriptive Study
RDT&E	Research, Development, Test and Evaluation (Cost)
SBD	Set-Based Design
TOFL	Take-Off Field Length
TRL	Technology Readiness Level
UQM	Uncertainty Quantification and Management

Latin symbols

A	Attribute value array	
A	Aspect-ratio (lifting surfaces)	
b	Span (lifting surfaces)	[m or ft]
B	Set of baseline (near-future) design options	
b_{root}	Span-wise length of root-insert (lifting surfaces)	[m or ft]
b_{tip}	Span-wise length of tip-extension (lifting surfaces)	[m or ft]
BCM	Body commonality matrix	
C	Set of constituent components of an aircraft	
C	Cost array	[USD]
c	Aircraft component	
c	Cost	[USD]
$\bar{c}$	Chord-length	[m or ft]
$\bar{c}_f$	Chord-length at location where the wing meets the fuselage	[m or ft]
CS	Common segment matrix	
F	Set of far-future design options	
H	Height	[m or ft]
L	Length	[m or ft]
$\mathbf{CR}^{(\tau)}$	Comparison matrix for components of type τ	
D	Design structure matrix (DSM)	
M	Mass array	[kg or lb]

m	Mass	[kg or lb]
n	Number of ...	
n	‘Difficulty’ power	
NCS	Non-similar segment matrix	
Q	Attached component array	
P	Power set	
R	Longest common subsequence recursive array	
r	Redesign effort	
S_1	Set of possible scenarios for near-future	
S_2	Set of possible scenarios for far-future	
s_1	Possible scenario for near-future	
s_2	Possible scenario for far-future	
S	Area	[m ² or ft ²]
SS	Segment list	
$\mathbf{T}^{(\tau)}$	Category-connection matrix for components of category τ	
u	Utility	
U	Utility array	
w	Weighting factor	
W	Width	[m or ft]
X'	Subsequence of X	

Greek Letters

α	Attribute value	
δ	Search length limiter for longest common subsequence problems	
ε	Error or tolerance	
Γ	Dihedral	[°]
Λ	Sweep	[°]
$\Lambda_{\bar{c}/4}$	Quarter-chord sweep	[°]
Ω	Commonality score	

Subscripts

ail	Aileron	
c	Cost	
cm	Common (component or segment)	

<i>ct</i>	Carry-through structure
<i>emp</i>	Empennage
<i>f</i>	Fuselage
<i>ht</i>	Horizontal tail
<i>m</i>	Mass
<i>nc</i>	Non-common (component or segment)
<i>tot</i>	Total
<i>vt</i>	Vertical tail
<i>w</i>	Wing
<i>wb</i>	Wing-box
<i>wf</i>	Wing-fuselage structures
<i>wfatt</i>	Wing-fuselage attachment
<i>wffr</i>	Wing-fuselage fairing
<i>fle</i>	Fixed leading-edge
<i>fte</i>	Fixed trailing-edge
<i>uc</i>	Undercarriage

Superscripts

- (*a*) Aircraft *a*, or component from aircraft *a*. Usually reserved for ‘baseline’ or ‘near-future’ entry-into-service aircraft.
- (*b*) Aircraft *b*, or component from aircraft *b*. Usually reserved for ‘descendant’ or ‘far-future’ entry-into-service aircraft.
- (*l*) Longer wing
- (*s*) Shorter wing
- (τ) Component category

To my parents
and
Shaun Schoeman

CHAPTER 1

Introduction

1.1 Betting the Company – Aircraft Evolvability and its Importance

The development of a brand-new aircraft is often an expensive and risky undertaking. In fact, when decision-makers at an aircraft manufacturer (referred to in this text as the ‘Original Equipment Manufacturer’, or ‘OEM’) elect to pursue a ‘clean-sheet’ design, they are often considered to be ‘betting the company’. Several examples serve to illustrate just how much trouble could be encountered when pursuing a new design. Recent ones include the Airbus A380 and Boeing’s 787 ‘Dreamliner’ development programmes. Each of these aircraft was revolutionary in some respect at the time they were introduced and both had experienced major cost overruns and time delays during their development (see, for example, Kingsley-Jones [1] for the A380 and Tang, Zimmerman, and Nelson [2] for the 787). Although sometimes necessary to remain competitive, this type of clean-sheet development programmes is rare, precisely because of the risks involved. OEMs are far more likely to devise only incremental advancements to existing designs. For such a strategy to be successful, however, the original (‘baseline’) design must be ‘evolvable’.

The evolvability (of an engineering system) is the extent to which the design of the system can be “inherited and changed across generations (over time)” [3]. This definition will be dissected and explored in detail in Chapter 2.

Note that the words ‘baseline’, ‘first-stage design’, or ‘near-future design’ is employed throughout this thesis to refer to the first-generation of designs, whereas the words ‘second-stage design’ or ‘far-future design’ refer to design options for subsequent generations. The words ‘descendent’ or ‘derivative’ are reserved for far-future designs which have an established relationship with a specific baseline. A near-future design indicates that it is being considered now for entry-into-service (EIS) in the near-future time-frame, whereas a far-future design is also studied at the present time, but for possible EIS further into the future. The combined near- and far-future design and objective spaces is collectively referred to as the ‘evolvability space’.

Evolvability is a type of change-related ‘ility’. Ilities are desired life-cycle properties of engineering systems, which are not intended to fulfil primary functional requirements [3], but which, nonetheless, often constitute much of the value of a system. It is therefore important for designers to recognise what ilities are required and to what extent these must be present in their products. Other change-related ilities include ‘flexibility’, ‘adaptability’, ‘versatility’, ‘robustness’, and the like. Along with evolvability, these are often grouped under a collective ility, called ‘changeability’ [4–6].

An evolvable baseline design may enable the manufacturer to reduce development cost and time significantly for descendants of this design. This is because it affords them the opportunity to reuse many design elements (such as segments of the airframe, components, systems, and development, manufacturing, and assembly processes), while upgrading only those that are necessary for subsequent generations to be competitive. The overall development cost and time of a descendant option will be referred to in this text as ‘redesign effort’ or ‘redesign cost’. An estimate of the redesign effort from a selected baseline design to different future design options is an indication of how evolvable this baseline is.

The practice of evolving existing designs is almost ubiquitous across the aircraft manufacturing industry. It can be seen often in military applications (for example, with the numerous airframe and avionics upgrades to the Lockheed Martin F-16 and F/A-18 [7]) and in commercial transport aircraft, with upgrades to propulsion, such as on the Airbus A320, A330, and Boeing 737 families, or ‘stretching’ of the fuselage for higher payload capacities. The Boeing 737 (Figure 1.1) is, in fact, an excellent example of a civil transport aircraft design that has undergone significant evolution and considerable discussion is devoted to its history in this thesis.

Furthermore, predicting the evolvability of design options is usually already considered even at the ‘conceptual design’ stage. Conceptual design is normally the first phase in the development of a new aircraft and is “where the designers look at



Figure 1.1: The Boeing 737 – an example of a civil transport aircraft design that has undergone significant evolution (Source: Boeing [8]).

a wide range of aircraft configuration concepts, perform trade studies of both the designs and the requirements, and ultimately settle on a single best design” [9]. It is a critical phase, as this is where designers have the most freedom (in terms of decision-making), but not much knowledge about the design [10, p. 19]. As the design progresses and decisions are made, knowledge gradually increases, but at the expense of design freedom. It becomes increasingly difficult (and expensive) to make changes and this is why the overall life-cycle cost is often largely determined by decisions made during conceptual design [10, p. 5]. It is therefore imperative that sound decisions are made at this stage.

OEMs are well aware of this and usually endeavour to study many design options before proceeding to the next phase (preliminary design) [9, pp. 13–17]. Generating and analysing large numbers of designs is made possible largely by the advent of modern computers, along with computational models that can swiftly estimate the aerodynamic, structural, propulsion, and weight properties of the concepts considered. ‘Generating’ a design option refers here to the process that accepts selected values for design parameters (usually geometry, payload, or performance constraint data) as input to the computational models, which are then executed to obtain the ‘attributes’ of the design option (usually performance, mass, and cost data) as output. This process of generating a design option will also be referred to as an ‘evaluation’ of a ‘point’ in the design space in the text. The process of searching for promising design options, among the thousands generated, is referred to as ‘design space exploration’. A detailed discussion on design space exploration is provided in Chapter 2.3.1.

Over the past two decades, several methods for exploring changeability and evolvability have been developed. These are discussed at length in Chapter 2.6. Although

much progress has been made in this field, including for evolvability, many challenges remain. Addressing some of the challenges is the purpose of this thesis. As will be discussed in the scope section, the work is limited to civil transport aircraft (commercial passenger airliners) and their airframes (i.e. fuselage, wing, empennage, pylon, and undercarriage structures, but not subsystems or propulsion). The challenges are described in the next section.

1.2 Motivation

In this thesis, two specific shortcomings regarding the current state-of-the-art for evolvability exploration during conceptual design are addressed. These were identified by means of a literature review (Chapter 2) and are as follows:

1. The computational cost with current methods could prove prohibitive when large numbers of designs are considered.
2. There appears to be a lack of systematic filtering processes to aid the designer in eliminating inferior design options for large evolvability studies.

These drawbacks are discussed in separate subsections below. Research questions are formulated for each, which were used to devise objectives for the research, as discussed in Section 1.3.

1.2.1 Computational Expense and Pre-Specification of Change Rules

With existing aircraft evolvability exploration techniques, descendant options are usually generated by applying pre-specified modifications to a selected baseline, such that each descendant in a study is linked only to a single baseline. One example from the literature comes from the exploration discussion in the thesis of Lim [7], in which Monte-Carlo sampling is employed to generate baseline-descendant pairs for a fighter aircraft evolution study.

Such studies are relatively straightforward to set up, as the ‘rules’ of changing the baseline into the descendant (e.g. increase span with a wing-tip extension, or increase fuselage length with a ‘plug’) are specified up front and the descendants can be generated by applying these rules. However, if large numbers of baselines and derivatives are considered, populating the near- and far-future design spaces could become prohibitively expensive in terms of computational cost.

To illustrate this problem, consider a hypothetical example with three baseline design options ($n = 3$), described by three corresponding design parameter vectors, and five far-future options ($m = 5$), described by five corresponding design

						Number of options
Near-future design options (baselines)			B1	B2	B3	$n = 3$
Far-future design options (possible descendants)			D1	D2	D3	$m = 5$

Evaluation: Mapping of parameter values, P , of a design option, (e.g. wingspan, wing area, etc.), to the attributes, A , of that design (e.g. range, cost, take-off field length, etc.). The mapping is performed by executing computational models, by means of an appropriate computational workflow, and is represented as $P \rightarrow A$.	Evaluation of near-future design options as clean sheets (3 evaluations) $P_{B1} \rightarrow A_{B1}$ $P_{B2} \rightarrow A_{B2}$ $P_{B3} \rightarrow A_{B3}$	Evaluation of far-future design options as clean sheets (5 evaluations) $P_{D1} \rightarrow A_{D1}$ $P_{D4} \rightarrow A_{D4}$ $P_{D2} \rightarrow A_{D2}$ $P_{D5} \rightarrow A_{D5}$ $P_{D3} \rightarrow A_{D3}$
	Evaluation of far-future design options as descendants of near-future designs (15 evaluations) $P_{B1} \cup P_{D1} \rightarrow A_{D1B1}$ $P_{B1} \cup P_{D2} \rightarrow A_{D2B1}$ $P_{B1} \cup P_{D3} \rightarrow A_{D3B1}$ $P_{B2} \cup P_{D1} \rightarrow A_{D1B2}$ $P_{B2} \cup P_{D2} \rightarrow A_{D2B2}$ $P_{B2} \cup P_{D3} \rightarrow A_{D3B2}$ $P_{B3} \cup P_{D1} \rightarrow A_{D1B3}$ $P_{B3} \cup P_{D2} \rightarrow A_{D2B3}$ $P_{B3} \cup P_{D3} \rightarrow A_{D3B3}$ $P_{B4} \cup P_{D1} \rightarrow A_{D1B4}$ $P_{B4} \cup P_{D2} \rightarrow A_{D2B4}$ $P_{B4} \cup P_{D3} \rightarrow A_{D3B4}$ $P_{B5} \cup P_{D1} \rightarrow A_{D1B5}$ $P_{B5} \cup P_{D2} \rightarrow A_{D2B5}$ $P_{B5} \cup P_{D3} \rightarrow A_{D3B5}$	

Total number of evaluations: $n + nm + m$ $= 3 + 15 + 5$ $= 23$

Figure 1.2: Example evolvability problem, demonstrating the number of evaluations needed to be performed to populate the design space.

parameter vectors (see Figure 1.2). Once the change-rules between the baselines and far-future options have been formulated, they are applied to populate the design space. This population, i.e. generating/evaluating all the designs, will involve computing the attributes of each option (which are calculated based on the values in the design parameter vectors), as well as the redesign effort, or redesign cost, between each baseline-descendant pair (which depends on the combined design parameter values of both the baseline and descendant, as well as the change rules). In this example, 3 baselines and 15 (i.e. $n \times m$) descendants would be generated, resulting in a total number of 18 evaluations. However, in addition to this, the far-future options have to be evaluated as clean-sheet designs as well (because developing a brand new design is also a valid option). This adds a further 5 evaluations, which brings the total to 23 evaluations.

Formally, if the time to evaluate a single design option is t_e and there are n baseline design options and m far-future design options, the total time to populate the near-future and far-future design spaces, using the current approach, can be calculated as:

$$t_{total} = (n + nm + m)t_e \quad (1.1)$$

In Equation 1.1, The first term in the brackets reflects the evaluation of the

baselines, the second term involves the evaluation of far-future designs as descendants of the baselines, and the term term is for evaluating the far-future options as *new* designs.

The 23 evaluations in the aforementioned example seems wasteful, as the values in each far-future design parameter vector are used five times to compute the attributes associated with that vector. What would be the implications of evaluating each far-future option only once, and then determining what the redesign effort between the baselines and far-future options would entail? This could provide a significant reduction in computational cost, as the total number of evaluations in the above example would be reduced from 23 to 8 (i.e. $n + m$ – the bottom block in Figure 1.2 is eliminated). If there were 1,000 baselines and 1,000 far-future options, and it took, say, 4 s to evaluate one design option, the evaluation cost is reduced from 46 days to 2.22 hours (if the redesign effort computing time is assumed to be negligible).

The first implication would be that the redesign effort would still have to be calculated for the baseline-descendant combinations. However, if the computational cost of estimating the redesign effort were substantially lower than evaluating a design point, the overall savings would still be significant. To demonstrate this, consider the following. If the time to estimate redesign effort is represented by t_r , then the total time to populate the near-future and far-future design spaces separately, and subsequently performing a redesign effort analysis, is given by:

$$t_{total} = (n + m)t_e + nmt_r \quad (1.2)$$

A plot, showing the results that normalised forms of Equations 1.1 and 1.2 provide, is shown in Figure 1.3. As can be seen, the savings in computational cost associated with the new approach could be considerable, if the time to estimate redesign effort is low.

Furthermore, it might not be necessary to compute the redesign effort for all the combinations. This is because it might be possible to first eliminate some baselines and far-future options based on other selection criteria.

The second implication of populating the near- and far-future design spaces separately, is that the possible effects on the performance of the descendant, due to reusing design elements, will not be accounted for. For example, if the wing design from a selected baseline is reused as-is on a descendant, it could have a stronger, and therefore heavier, structure than needed. Inevitably, this would lead to a reduction in the achievable range and increase the field-length requirements (i.e. the reuse has adverse effects on the performance). The performance attribute values

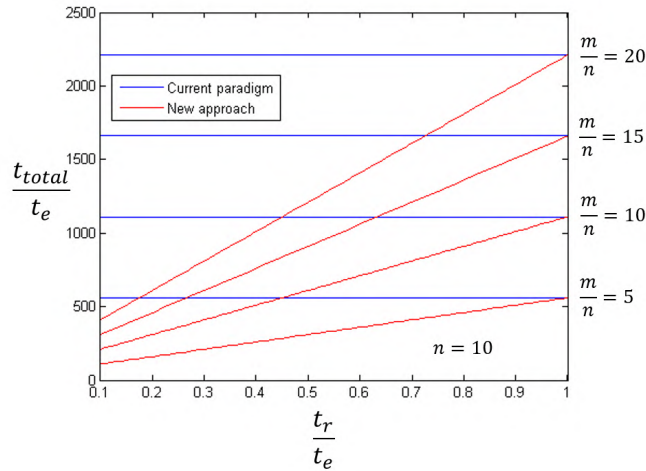


Figure 1.3: Current evolvability assessment computation time and that of the proposed approach.

calculated would therefore be different if the far-future were a clean sheet design or a descendant. OEMs do, however, often make the structures of descendant aircraft lighter if necessary, by decreasing the material thickness (gauge).

To investigate the differences in mass and performance between a clean-sheet design and one that has commonality with a selected baseline, the author performed a simple test. An extreme case would be where the whole airframe is common with a previous baseline that was sized based on earlier technology. Therefore, a baseline aircraft was modelled in the NASA Flight Optimisation System (FLOPS), an aircraft sizing and optimisation program (see McCullers [11]). The baseline selected was the Airbus A320 and FLOPS was calibrated using publicly available data for this aircraft (see Chapter 6.2.4 for more on this calibration). A new engine that provides a 10% reduction in fuel-burn (typical for generational advances) was then incorporated, after which FLOPS was executed twice using this new engine with the same airframe dimensions. For the first run, FLOPS was allowed to compute new structural weights, based on the improvements that the new engine provides, whereas in the second run it was constrained to use the baseline weights. The first run therefore constitutes sizing of a new aircraft, whereas the second run is a for a derivative with only an engine change. The results are shown in Table 1.1. As can be seen, the differences in weights and performance are all below two percent¹.

For the purposes of this research, these effects on weight and performance were assumed to be small enough and were therefore ignored. However, it is advised that, before applying the techniques in this thesis to a specific problem, a test

¹This difference will probably be higher for longer-range aircraft.

Table 1.1: Mass and performance effects of commonality.

	Baseline	New	Derivative	% Difference between derivative and new
Operating empty weight [lb]	91,144	90,891	92,502	1.74
Gross weight [lb]	162,507	157,974.6	159,855.6	1.18
Mission fuel [lb]	40,613.6	36,333.9	36,603.9	0.74
FAR Takeoff Field Length [ft]	6,126	5,955	6,026	1.18
FAR Landing Field Length [ft]	6,227	6,123	6,166	0.70

such as this is always performed to confirm whether the impact of commonality on other attributes of interest are negligible (i.e. to confirm that commonality and other attributes of interest are reasonably independent). Note that the attribute of cost is definitely not independent of commonality and is heavily affected by redesign effort. The calculation of cost therefore forms part of the redesign effort estimation. For the techniques in this thesis to be fully ready for use in practice, methods to account for the effects of commonality on all the attributes will have to be developed. Some suggestions for such methods are provided in the future work section in the conclusion of this thesis.

Therefore, if 1) the computational cost of estimating redesign effort is low in comparison with design point evaluation cost and 2) the attributes (excluding cost and commonality itself) of the far-future options are assumed not to be significantly affected, based on whether they are clean-sheets or descendants, the baseline and far-future design space could be sampled separately, and redesign effort calculated afterwards. As shown above, this could lead to a significant reduction in overall computational cost.

Should this approach be adopted, the question arises of how the redesign effort should be determined. For example, if a descendant has a larger wing area than the baseline, how was this achieved? Was it by way of a chord increasing ‘plug’, a wing-tip extension, a ‘root-insert’, or a combination of some or all of these? This problem arises because such modifications were not applied to a specific selected baseline. Obviously, a pre-specified rule could still have been dictated, such as “if there is an increase in wing area, this is always achieved through adding a tip-extension and a chord increase plug”. However, this is problematic for several reasons, as listed below:

- The specified rule might not be a good, or even feasible, approach to achieving the change. For example, considering again the wing area, there is a limit to how much chord can be increased. It might be preferable to add a root-insert to achieve the change. Therefore, the designer would probably have to specify a number of rules and determine the one that produces the best results. However, this would still not answer the question, for example, of how large the wing-tip extension, root-insert, and the like, would need to be.
- If there were multiple different configuration concepts, change rules would probably have to be specified for each combination of configurations considered.
- Specifying the change rules manually, particularly when multiple configuration concepts are considered, would become a time-consuming task that would leave less time available for actual exploration.

The points above alluding to different ‘configuration concepts’ formed a particularly important motivating factor for this research. This is because future aircraft might need to encompass more unconventional configurations, for the aviation industry to meet the challenging targets regarding environmental friendliness [12]. It is therefore essential to study how current designs could evolve into such unconventional configurations. Configuration here refers not only to how the major components (the fuselage, wing, empennage surfaces, propulsion, and undercarriage) are arranged, but also to their type (e.g. for propulsion – turbofan, open rotor, hybrid-electric, and the like). The ‘conventional’ configuration in this text refers to the current dominant ‘tube-and wing’ layout of modern commercial aircraft, as embodied, for example, by the Boeing 737 (Figure 1.1). Figure 1.4 shows examples of civil transport aircraft designs that have novel configurations.

Therefore, to make the expected computational cost savings of sampling the near- and far-future design spaces separately feasible, a technique/s is/are needed that could estimate the redesign effort between two given design options automatically. In addition, as discussed, these techniques must be able to estimate the redesign effort across two designs that have dissimilar configurations.

Intuitively, redesign effort is strongly dependent (amongst other factors) on the degree of ‘dissimilarity’ between two airframes. The more dissimilar they are, the more difficult it would be to change the one into the other, and the higher the redesign effort would be. How does one measure dissimilarity – or similarity? One way of measuring similarity is by assessing the ‘commonality’ (of components, airframe segments, and processes) across the two designs. Commonality was employed in this research as a measure of redesign effort and is introduced in more detail in Chapter 2.6.3.3.



Figure 1.4: Novel aircraft configuration concepts (source: NASA [13]).

Apart from the potential savings in computational cost, another, related, opportunity was identified. If the change rules were not to be pre-specified any longer, the question arises whether it would be possible to obtain suggestions for change rules from the exploration study itself, rather than the designer having to specify them. This could enable new types of evolvability exploration activities, such as selecting a far-future option with a novel configuration and subsequently determining how a selected baseline option could evolve to it. Interactive design space exploration could be greatly enhanced using such techniques if implemented in a software tool. For example, the user could select a baseline and descendant pair and swiftly obtain a visual representation of the redesign effort and possible commonality. To obtain such suggestions for change rules would, again, require some means of automatic similarity assessment between designs, particularly if this similarity assessment could indicate which components, or airframe segments from the baseline could be reused (i.e. made common) in the new design.

The need for techniques to assess similarity and estimate redesign effort can be stated in terms of a research question, as follows:

Research Question 1: How can similarity between airframes be assessed automatically to predict redesign effort?

This research question will be translated into an objective in Section 1.3. First, however, a discussion on the second shortcoming of current evolvability exploration techniques is provided.

1.2.2 Lack of Systematic Filtering Procedures

The second shortcoming of current evolvability (and changeability) exploration methods is that there appears to be an absence of techniques to systematically filter out less desirable design options. Many techniques exist that focus on finding ‘good’ designs (see, for example, Ref [14]). However, at the conceptual design stage, it is often important to refrain from converging to a single design prematurely. In other words, it is desirable to keep the design space open and delay decisions, to facilitate acquiring further knowledge before committing to a single design too early. This is because uncertainty is high, and an apparently good design could prove to be infeasible as new knowledge is obtained, which could lead to costly iterations [15, 16].

The issue of converging to a single design prematurely has long been appreciated, and a specific design approach was developed by certain companies, such as the Toyota Motor Corporation [17] and academia to address this problem. This approach is called ‘Set-Based Design’ (SBD), or Set-Based Concurrent Engineering. In SBD, several designs are developed in parallel and, as knowledge increases, this set of designs is systematically pruned of demonstrably inferior options. Several criteria have been proposed to filter out designs in this manner and are described in Chapter 2.8. Unfortunately, these techniques are not directly usable for the multiple-timeframe design spaces of evolvability studies, as will be shown in the literature review. The second research question is therefore formulated to address this deficiency. It is stated as follows:

Research Question 2: How can set-based design methods be devised and/or adapted for evolvability exploration studies?

This question is translated into an objective for the research in Section 1.3, which follows.

1.3 Aim and Objectives

According to Blessing and Chakrabarti [18], to be successful, engineering design research must be academically and practically worthwhile, as well as realistic. Academically worthwhile means that the research is “sufficiently challenging and generic and its solution is expected to contribute to our knowledge and understanding” [18]. Practically worthwhile means that the techniques devised as a result of the research are sufficiently beneficial to the practitioners for which they are prescribed (aircraft conceptual designers in this case). Finally, ‘realistic’ simply means that the research can be performed within the time and resources available.

It will be shown in this section that the aim and objectives, if achieved, adhere to these criteria.

The aim was formulated as follows:

Aim: To develop computational techniques that would enable aircraft designers to explore the evolvability of their designs more efficiently and effectively during the conceptual design stage.

‘Computational techniques’ refer to algorithms, filtering criteria, object models, or anything else that could be implemented in a software tool to support evolvability exploration. ‘Efficiency’ refers to the number of design options and configurations that could be studied and down-selected within a given time. Effectiveness, on the other hand, is the likelihood of finding design solutions that have good evolvability characteristics (i.e. meet or exceed near-future requirements and have many possible derivatives that meet or exceed future requirements).

Achieving this aim is *academically worthwhile*, because achieving it is expected to increase our understanding of the exploration of the evolvability of engineering systems in general, and aircraft in particular. It is challenging, because, as will be shown, complex methods from the different fields of mathematics, systems engineering, computer science, and aircraft design had to be adapted and combined to successfully devise the techniques proposed. Achieving this aim is also expected to increase the quality of evolvability-related decisions made. This could lead to more evolvable baseline designs, which could, in turn, reduce the development risk of far-future designs, while still ensuring their competitiveness. Pursuing the aim is therefore *practically worthwhile*. In order to make the research *realistic* it was scoped appropriately, as discussed in Section 1.4.

The first objective was formulated as follows:

Objective 1: To devise computational techniques that could assess similarity between airframes automatically.

As can be seen, Objective 1 is a reformulation of Research Question 1. Pursuing this objective is *academically worthwhile*, because, as will be shown (and stated in the aim description), it involves the application, combination, and extension of existing techniques from mathematics, systems engineering, computer science, and aircraft design. Achieving this objective is both challenging and increases knowledge, as it demonstrates how these techniques could be applied in a new domain. It is practically worthwhile, as it is expected to lower the computational cost associated with evolvability studies, which would allow for more efficient and effective exploration. Again, the work was sufficiently scoped to make it *realistic*.

The second objective was formulated as follows:

Objective 2: To devise new and/or adapt existing set-based design criteria to enable systematic filtering of the evolvability design space.

Again, Objective 2 is a reformulation of Research Question 2. This objective is *academically worthwhile*, as it involves the extension of existing techniques to a more complex problem. It is *practically worthwhile*, because it will provide designers with a means to systematically reduce the thousands of designs they consider, thereby promoting efficiency. Finally, the work for this objective was also scoped to make it *realistic*.

1.4 Scope

The scope in this research was limited to studying and advancing the evolvability of civil transport aeroplanes (i.e. aircraft that would typically be certified in accordance with the CS-25 [19] or FAR-25 [20] regulations). The focus is also (to an extent) more on single-aisle transports, but examples are provided for widebodies. Notwithstanding the concentration on civil transports, the techniques developed should be applicable, or easily adaptable, to business, general aviation, or military aeroplanes. A similar approach as proposed in this research could possibly also be adopted for other complex engineering systems. The reason for focusing on civil aeroplanes was simply to make the scope of the work manageable and because the author had access to experts in the design of this type of aircraft.

As will be discussed in Chapter 2.4.2, a concept closely related with evolvability is that of product families. For this research, the aircraft members of a family are considered to be pre-planned from the outset of the design of the main member. With evolvability, this is not necessarily the case. The techniques developed in this research should be easily applicable to aircraft family design.

Furthermore, the scope was limited to the conceptual design stage, because, as already stated, this is the phase where decisions are made that affect the majority of the overall life-cycle cost of an aircraft programme. It is therefore believed that focusing on aiding decision-making during this phase could have the largest positive effect on the overall value of aircraft programmes. The work may, in some instances, be applicable to later stages of the design process, but will then likely be only appropriate for the design of subsystems and be employed within tighter constraints.

The study of the evolvability of subsystems (e.g. flight control, environmental control, hydraulics, etc.) and propulsion was not considered (although airframe-propulsion integration was indeed taken into account). Again, this decision was made to ensure that the research could be completed within the given timeframe. An additional reason for not considering subsystems was that conceptual studies would usually begin with investigating the airframe and propulsion only. However, subsystems are being considered increasingly early in design, and future work will need to consider the evolvability of the combination of airframe and systems. Note that salient aspects relating to geometry, for example fuel tank volume (which is an important constraint when considering evolvability), are implicitly covered by the techniques devised in Chapter 4.

Finally, it is important to note that the research discussed in this thesis was not a study to determine which current and future aircraft configurations are/would be the most evolvable. Rather, it was intended to produce tools that could aid designers in identifying evolvable designs. Therefore, the results from demonstrative examples that, for instance, indicate that a certain configuration is more evolvable than another should be interpreted bearing in mind the limitations of the aircraft sizing and performance models employed. Furthermore, the purpose was not to study trends in future technologies and the environmental, and socio-economic, and political factors that could influence future aircraft design. Accordingly, it is assumed that relevant information is available as input to the exploration studies. That being said, however, the techniques could aid in, e.g. developing technology roadmaps, as they could help determine which technologies could be incorporated more easily in the design options under consideration, along with their effect on performance, cost, and the like.

1.5 Research Methodology

Because the work introduced in this thesis is a form of prescriptive engineering design research, a four-step methodology for the research was formulated by adapting the generic engineering design research methodology (DRM) proposed by Blessing and Chakrabarti [18]. Using the classification of engineering design research, as presented by Blessing and Chakrabarti, the research described in this thesis could be considered a ‘Type 3’ study. In a Type 3 study, the majority of the work focuses on producing the design support tool. This type was chosen because it was anticipated that only a preliminary evaluation would be possible within the given timeframe. The research methodology followed is shown in Figure 1.5. A description of each step is provided in the sections that follow. Note that this approach was iterative and that many of the steps were repeated to refine the

work.

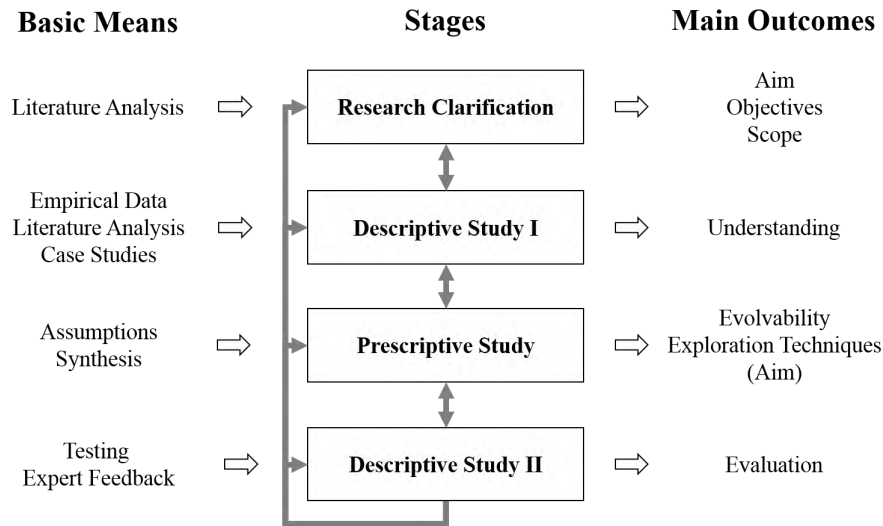


Figure 1.5: Research methodology adopted (adapted from Blessing and Chakrabarti [18]).

1.5.1 Research Clarification

In this step, the literature was consulted to gain an initial understanding of the phenomenon of the evolvability of engineering systems and to determine the industrial need pertaining to aircraft evolvability exploration, in particular. The purpose was to identify important challenges and opportunities regarding evolvability exploration at the conceptual design stage. The outcome of this step was the aim and objectives (that have to be academically and practically worthwhile, as well as realistic), as stated and discussed in Section 1.3.

1.5.2 Descriptive Study I

The purpose of this step was to ‘describe’ the phenomenon of evolvability of engineering systems and, especially, aircraft. The objective was to gain an improved understanding of the challenges identified in the previous step, to determine what others have done to address these challenges, and to identify possible methods to help answer the research questions.

Several methods were employed during the execution of this step. The first was to consult the academic and industry literature. Secondly, the author took part in discussions with conceptual design experts at Airbus UK. Finally, to develop techniques to estimate redesign effort, requires knowledge of which changes are actually possible and reasonable. To acquire this knowledge, several case studies

were performed on existing aircraft programmes, for which the literature was the main source.

1.5.3 Prescriptive Study

In this step, computational techniques were developed, pursuant to the objectives. Several assumptions were made, which were vetted by design experts at Airbus UK. These are stated clearly throughout the thesis, along with their justification. The synthesis itself involved creating algorithms and filtering criteria, by adapting and extending concepts and methods from (as already mentioned) mathematics, computer science, systems engineering, and aircraft design. These were implemented subsequently into software for testing purposes (the next step).

1.5.4 Descriptive Study II

The purpose of this step was to describe the effects of applying the prescribed techniques to the problem identified. The objective was therefore to evaluate the work. Evaluation of prescribed techniques in engineering design research usually involves three different types of evaluation [18], namely ‘support’, ‘application’, and ‘success’.

The techniques are verified in the support evaluation, i.e. it is determined whether they produce expected results and whether these results are consistent. In the current research, the support evaluation was conducted by applying the proposed techniques to several designs used in existing aircraft programmes, in order to determine whether they produced results consistent with what had been achieved actually in these programmes. The techniques were also applied to a hypothetical study involving future aircraft designs.

Employing the application evaluation facilitates determining whether implementing the techniques was likely to achieve the aim of the research. In other words, in the current research, it had to be determined whether the techniques would actually improve the efficiency and effectiveness of evolvability exploration for civil aircraft at the conceptual design stage. Because it is impractical to test the proposed work in an actual industrial setting (such a study would be extremely time-consuming, involve risk on the part of the OEM, and would require access to proprietary data), only an initial application evaluation was performed. This involved demonstrating the techniques to aircraft conceptual design experts and obtaining feedback from them.

Success evaluation is more general. Here, it is determined whether the proposed techniques would provide value and is expected to be useful in practice. Again,

for this research, only an initial success evaluation was conducted, by interviewing design experts.

1.6 Thesis Structure

The structure of the thesis is shown in Figure 1.6, which shows that the content of each chapter (with the exception of Chapter 7) relates to the research methodology steps discussed in the previous section.

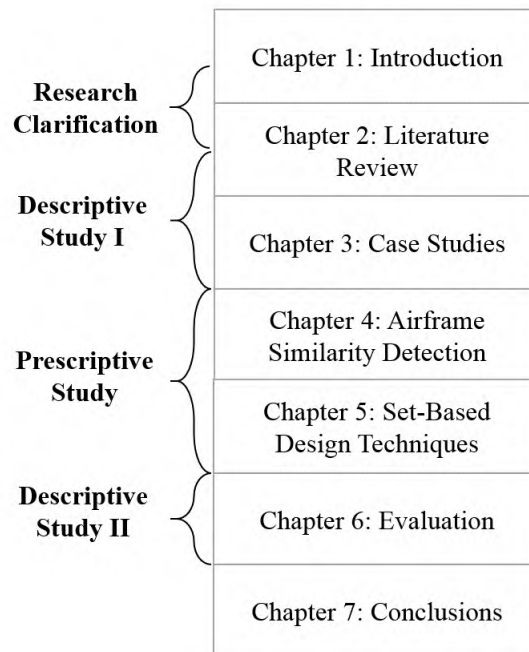


Figure 1.6: Structure of the thesis.

This introduction states the general area of research, explains the motivation and provides the aim and objectives (Research Clarification). It is followed by Chapter 2, in which i) important definitions and concepts are introduced and ii) the limitations of current approaches and associated opportunities are highlighted. Chapter 2 therefore reports further on the Research Clarification (i.e. it explains how the aim and objectives were formulated), as well as the initial Descriptive Study I (the challenges in evolvability exploration are described in more detail, and possible solutions are identified).

Chapter 3 contains several case studies in existing civil transport aircraft programmes in which evolvability was exploited. The purpose was to identify which type of design changes are generally possible and reasonable, as well as which type of design changes are usually more difficult to implement than others. Its contents therefore serves to further report on the Descriptive Study I, because it increases

understanding of the phenomenon of aircraft evolvability. The work presented in Chapter 3 resulted in a number of observations and constitutes an additional contribution of this research.

In Chapters 4 and 5 the actual prescribed techniques are proposed (Prescriptive Study). In Chapter 4, a computational framework is introduced, pursuant to Objective 1, that consists of methods to assess similarity across airframes automatically. Proposed techniques for applying set-based design to evolvability studies (pursuant to Objective 2) are introduced in Chapter 5.

The purpose of Chapter 6 is to report on the evaluation of the proposed techniques (Descriptive Study II). The application of the techniques is demonstrated, followed by a discussion on the feedback obtained from a panel of aircraft conceptual design experts.

Finally, the conclusion of the thesis is presented in Chapter 7. The contributions are highlighted and avenues for future research are discussed at length. The conclusion is followed by appendices that were included in support of the different chapters.

CHAPTER 2

Literature Review

Chapter Summary: The research presented in this thesis involves concepts from various fields. The purpose of the review presented here is therefore to introduce these to the reader and to highlight the limitations and opportunities related to the existing state-of-the art in evolvability exploration during aircraft conceptual design. Brief overviews of aircraft design and computational approaches in aerospace design are provided first, followed by an introduction to the concept of evolvability in engineering systems, as well as some examples of evolution in actual aircraft. After this, an investigation into existing procedures for designing for evolvability in engineering systems is provided. The findings of this investigation include that there is a need for i) automatic similarity assessment techniques that could predict commonality across aircraft of different configurations to expedite tradespace exploration; and ii) methods that could aid in systematically narrowing down the design space, while, at the same time, ensuring that design freedom is maintained later into the design process. Promising approaches identified for point (i) include design-structure matrix-based methods, optimisation, and expressing fuselage and nacelle commonality prediction as a longest-common subsequence problem. Techniques from the field of set-based design are identified and discussed as a means to address point (ii).

2.1 Introduction

The research presented in this thesis involves concepts from various fields of enquiry. These include: aircraft conceptual design, artificial intelligence, decision theory, dynamic programming, and the study of complex engineering systems, amongst others. The purpose of this literature is therefore twofold: i) to introduce the reader to the many concepts involved; and ii) to highlight the limitations and associated opportunities identified with the existing approaches to evolvability exploration.

The discussion starts with an overview of aircraft design (Section 2.2) and computational approaches in aerospace conceptual design (Section 2.3). After this, the concept of evolvability in engineering systems is introduced (Section 2.4). This is followed by Section 2.5, which covers the phenomenon of evolvability in aircraft. In Section 2.6, existing techniques for designing for and exploring evolvability are introduced and their limitations are highlighted. Finally, two approaches that seem promising for addressing the limitations, computational similarity assessment techniques and set-based design, are presented in Sections 2.7 and 2.8, respectively.

2.2 Aircraft Design

2.2.1 Overview of the Aircraft Design Process

According to the United States Accreditation Board for Engineering and Technology [21], engineering design is “a process of devising a system, component, or process to meet desired needs and specifications within constraints”. It can be a highly intricate and expensive undertaking, especially when the system to be designed is an aerospace vehicle. Keane and Nair [22] even state that “aerospace engineering design is one of the most complex activities carried out by mankind”. Subsequently, many models of the design process for aircraft (and other aerospace systems) exist. One common way to break down and generalise the process is to divide it into three major consecutive (but strongly overlapping) phases. These are the ‘conceptual’, ‘preliminary’, and ‘detail’ design phases, and are depicted in Figure 2.1, together with the major considerations involved in each.

During conceptual design, several thousand designs are usually investigated and trade studies are performed of the designs and requirements [9, p. 13]. The result is usually a single ‘best’ design. The conceptual design phase is the main focus of this thesis and will be returned to shortly.

After conceptual design, the preferred design is taken forward to the preliminary

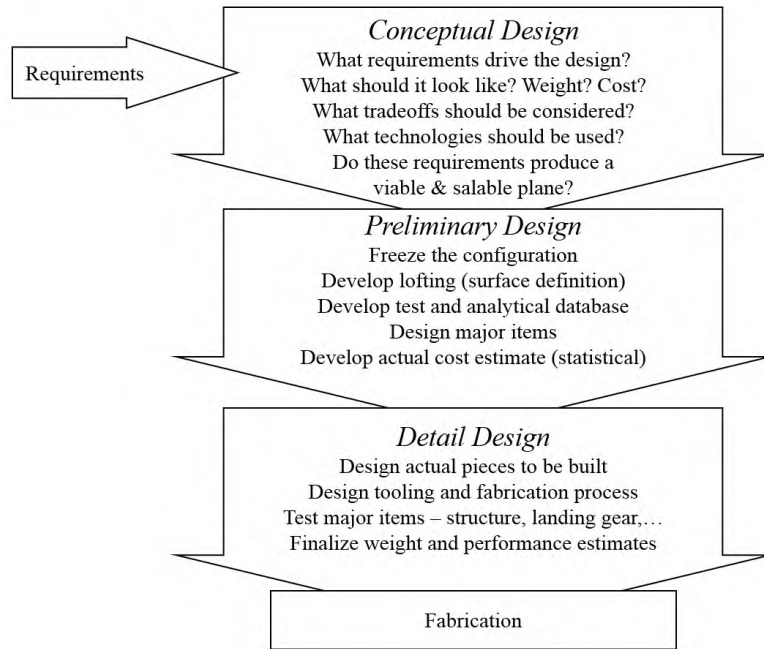


Figure 2.1: The phases of aircraft design
 (adapted from Raymer [9, p. 13]).

phase. Here, it is refined and developed to a such a level of detail that the company can decide on whether to commit to it [9, p. 14]. If the decision is indeed to continue, the individual parts are designed in the detail design phase, after which they are manufactured and assembled to create the final product.

2.2.2 The Conceptual Design Phase

During conceptual design, thousands of designs options (of several configurations) are usually generated and analysed. A combination of methods is employed, including textbook (see for example references [23], [9], [24], [25]) and computational methods, such as Multi-Disciplinary Design Optimization (MDO). Numerous configuration concepts are also often explored, to find which of these would best match the requirements, while being “technically feasible” and “commercially viable” [26, p. 14][25, p. 10]. ‘Configuration’ here refers to the general layout of the major components of the aircraft, i.e. the fuselage, wing, undercarriage, empennage surfaces, and engines. Along with defining the overall airframe geometry and the location, number, and type of engines, initial decisions regarding the cabin internal arrangement, systems, materials, and manufacturing processes are made as well [26, p. 14]. The degree to which novel technologies would be exploited by the new aircraft is usually also considered at this stage. To facilitate decision-making, preliminary estimates for performance are made, and three-dimensional electronic drawings are produced, along with weight and balance diagrams [25, p. 10].

The quantity of data produced for each design is normally somewhat limited, though (i.e. the level of detail of the definition is relatively low) [26, p. 14][9, p. 14][22, p. 20]. This is because exploring each concept in depth is usually not practical. However, making sound decisions at this stage is essential, because most of the cost of a product is usually determined by choices made during conceptual design [10, p. 5]. This is also the phase in which the designer has the most freedom, but where knowledge (about the ultimate solution) is limited [10].

Modern computers have greatly facilitated increasing this quantity of data, as well as the number of design options that could be generated and analysed during conceptual design. Subsequently, many design and decision-making techniques that exploit the power of computers have been developed and are broadly discussed next.

2.3 Computational Approaches

The power of modern computers has revolutionised the aerospace sector. Their use in the modelling and simulation of aerospace systems has had a particularly substantial impact, enabling designers to perform studies of unprecedented detail [22]. Computers are employed throughout the design process, and their use in computer-aided design (CAD), computational solid and fluid mechanics (CSM and CFD), and multi-disciplinary design optimisation (MDO) have been particularly successful. For a comprehensive overview of computational methods in aerospace design, the reader is referred to Keane and Nair [22].

Conceptual design activities that have been enabled or enhanced by modern computational power include: design search and optimisation (see Refs [25][pp. 4-5], [22, pp. 45-51], and [27] for overviews of optimisation in design), design space exploration (see Ref. [28]), parametric geometry modelling (e.g. Refs [29], [30], and [22, pp. 55-74]), uncertainty quantification and management (e.g. Refs[31] and [32]), set-based design [15], and many others. The activities of special relevance in this research are discussed in more detail in the following subsections.

2.3.1 Design Space Exploration

The ‘design space’ (DS) is defined in this research as a hypercube confined by the specified valid ranges of all the design variables considered. This is the same definition as in Guenov et al. [33]. Note that the DS could also have discrete variables, where, for example, ‘engine type’ is represented by an integer value. A DS usually has an associated ‘performance’, ‘solution’, or ‘objective’ space. Each design point in the design space (i.e. a combination of input values for the

design variables) is mapped to a point in the performance space by means of a ‘computational workflow’.

According to Kang, Jackson, and Schulte [34], ‘Design Space Exploration’ (DSE) is the “activity of discovering and evaluating design alternatives during system development”. It is used to support decision-making in design and especially to increase understanding of the trade-offs involved in a design problem; to perform sizing, design change, and ‘what-if’ studies [28]; and – ultimately – to find the most promising solutions.

In line with the scope of the research, DSE is considered here in the context of computational conceptual design. Within this context, the aircraft is described by numerous quantitative models that relate to the parametric geometry (see 2.3.2), performance, and cost, amongst others [28]. These models are assembled in a computational workflow that computes outputs based on selected input values. An illustration of a computational workflow for aircraft sizing is provided in Figure 2.2. This rendering was produced by AirCADia Explorer, an interactive conceptual design space exploration tool developed by the Advanced Engineering Design Group at Cranfield University [35]. In this workflow, input variables are represented by green ovals, whereas output (or intermediate) variables are shown in red. The computational models are represented by the purple rectangles. By executing the workflow for different combinations of input values (called sampling), the design and performance spaces are populated with design options which can then be evaluated. Note that several methods exist that enable computational workflows to be reconfigured, such that selected outputs could be changed to inputs and vice versa (see for example [28]).

The DS could be sampled and analysed in several different ways. Strategies include ‘Design of Experiments’ (DoE) techniques, optimisation, and sensitivity and isocontour analyses [28], along with Set-Based Design (SBD). The techniques proposed in Chapter 4 could be used with any of these strategies, but DoE was used specifically to demonstrate these techniques (as presented in Chapter 6). DoE refers to systematic application of several statistical methods to study the effects of multiple input variables on the output variables of a particular system (Montgomery [36, p. 1 and 11]). The statistical sampling strategies of DoE are useful in design space exploration (amongst many other applications) and of particular interest here. Many sampling strategies exist, but one of the most common and intuitive (though most computationally expensive) is the full-factorial strategy [37, p. 16]. In sampling, the input variables (also called factors) are discretised into different ‘levels’ (the number of values that the variable could assume) [37, p. 14]. In a full factorial study, the samples are given by the combination of all the levels

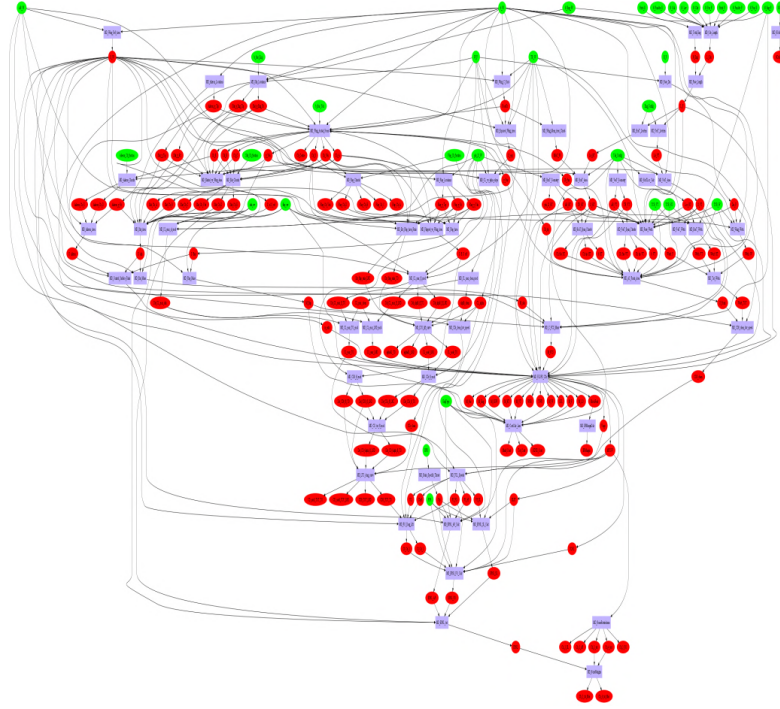


Figure 2.2: Example of a simple aircraft sizing workflow (created by the author using AirCADia Explorer [35]).

of all the factors considered [37, p. 16]. Full factorial sampling was used in this research because of its simplicity, but any strategy could be used.

Another design space exploration technique of particular importance in this research is Set-Based Design (SBD), to which the work presented in Chapter 5 specifically contributes. SBD is an approach where, instead of selecting a limited number of ‘optimal’ design solutions, ‘inferior’ solutions are removed systematically from the design space. SBD is discussed in depth in Section 2.8.

Finally, dedicated exploration frameworks have been developed for engineering systems that undergo change (including evolvable systems). Amongst these are ‘Multi-Attribute Tradespace Exploration’ (MATE) and associated concepts. Because MATE is an approach for which several exploration techniques exist, it was selected as the main exploration framework to demonstrate the proposed techniques of the research. This is discussed in more detail in Section 2.6.4.3.

2.3.2 Geometry Modelling and Parameterisation

Modelling geometry is a fundamental activity in aerospace design [22] and, because any assessment of commonality across aircraft would naturally involve identifying similar geometric features, it is also particularly relevant to this research.

The mathematical representation employed to represent the geometry could have

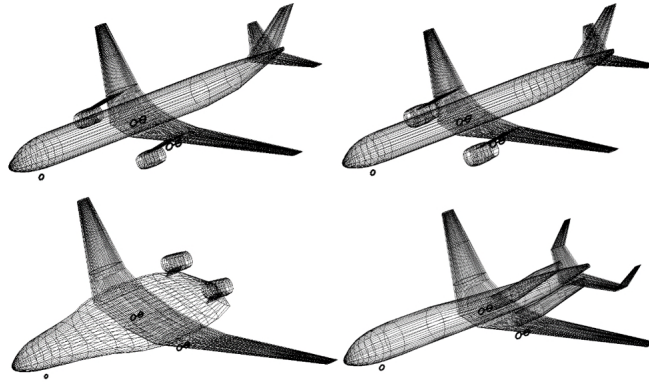


Figure 2.3: Geometries created by the author using the class-function/shape-function transformation representation by Kulfan [29].

a significant effect on computational resources [30]. It is especially desirable to limit the number of design variables describing the geometry [30]. This is normally done through parameterisation of the geometry of the constituent components that make up the vehicle. Many approaches for geometry parameterisation exist (see [22, pp. 55-74] for an overview), but for this research the ‘class-function/shape-function transformation’ (CST) representation proposed by Kulfan [29], [30] was employed. This is because of the advantages associated with the class function representation and the fact that it was already implemented in AirCADia. In CST, the class functions represent classes of aerofoils, axisymmetric bodies, and axisymmetric nacelle geometries, whereas the shape function enables the definition of unique geometries within each class [29, p. 157]. Examples of the geometries that could swiftly be created using this representation are depicted in Figure 2.3.

2.3.3 Knowledge-Based Engineering

Knowledge-based engineering (KBE) comprises technology that can significantly automate some of the activities in computational design and thereby reduce product development cost and duration. The study of KBE technology is at the intersection of diverse disciplines, such as Artificial Intelligence (AI), Computer Aided Design (CAD), and computer programming [38, p. 55]. La Rocca [39, p. 161], [38, p. 57], provides the following definition for KBE:

Knowledge-based engineering (KBE) is a technology based on the use of dedicated software tools called KBE systems, which are able to capture and systematically reuse product and process engineering knowledge, with the final goal of reducing time and costs of product development by means of the following:

- Automation of repetitive and non-creative tasks.

- Support of multidisciplinary design optimization in all the phases of the design process.

A KBE system makes use of a ‘knowledge base’ (KB) to solve domain-specific problems via an ‘inference mechanism’ [39, p. 58]. The KB is a repository in which the knowledge is stored, while the inference mechanism is a type of reasoning that can be performed on the basis of the specific input for the problem, as well as the knowledge in the KB [39, p. 58]. To implement the inference mechanism in an application, it has to be structured and formalised into a ‘knowledge representation’ [39, p. 161]. There are two common types of representation systems, namely Rule-Based Systems (RBSs) and Frame-Based Systems (FBSs). An RBS usually consists of a series of IF-THEN rules, while an FBS is usually an object model that contains the necessary knowledge to solve a specific type of engineering problem. FBSs are therefore usually developed in line with the object-oriented programming paradigm [39, p. 161], [40] and are generally more suitable to engineering design [39, p. 161].

One of the distinguishing features of KBEs is that their programming language could be used to drive a CAD engine [39, p. 161]. For conceptual design, the CAD engine often incorporates parametric geometry techniques, such as those discussed in Section 2.3.2 (see La Rocca [38] for an in-depth discussion on this topic).

As will be explained in Chapter 4, the proposed similarity assessment techniques collectively form an inference mechanism for a frame-based KBE system. A notional knowledge representation for this system – a generic aircraft object model – is also provided in Chapter 4. However, note that developing a fully fledged KBE system was beyond the scope of this research and the techniques were developed mainly in MATLAB[®] and not a dedicated KBE language.

2.4 Engineering Systems Evolvability

Before proceeding to the definition for evolvability, a discussion regarding the concepts of ‘change’ and ‘changeability’ in ‘engineering systems’ is required. The systems referred to in this context are ‘engineering systems’, which, according to the MIT Engineering Systems Division’s 2008 strategic report [41], are:

A class of systems characterized by a high degree of technical complexity, social intricacy, and elaborate processes, aimed at fulfilling important functions in society.

This definition applies readily to modern civil transport aircraft, as these take

years to design and develop, involve many thousands of people in both their development and operation, and fulfil the important function of providing relatively fast transportation for millions of passengers around the world, which extensively supports the global economy.

2.4.1 Engineering Change

Ross et al. [5] define change as “the transition over time of a system to an altered state”. According to them, each change event has three aspects, namely an ‘agent of change’, a ‘mechanism of change’ and an ‘effect of change’.

The change agent is the initiator of the change event and can be internal or external to the system. Note that a cause (or driver) of change is not necessarily the same as the change agent. The change agent actually performs the change, but the cause/driver refers to that which drove the change agent to initiate the change in the first place. In the case of aircraft evolvability, the change agent is usually the company that designs and manufactures the aircraft. The change drivers applicable to evolvability are changes in needs and requirements, which could arise as a result of a ‘changing world’ (e.g. evolving technology, changing socioeconomic factors, the actions of the competition, changing customer preferences [airlines and passengers], and so forth). The reader is referred to Fricke et al. [42], Wenzel et al. [43], and Schulz and Fricke [44] for a more general treatment of change drivers.

The mechanism of change (also called the change rule) is the ‘path’ from one state to another and involves activities such as [45] redesign (especially relevant to evolvability); substitution, addition, or subtraction of components/parts; and modification (where a parameter of a component is changed). Note that there is a ‘cost’ or resource expenditure associated with the mechanism.

Change effect pertains to intended and unintended consequences of applying the change mechanism to the system. Ross et al. (2007) distinguish between three types of change effects. The first is simply no effect, in which the system is referred to as ‘robust’. The second effect is scaling, which happens when there is a change in the “parameter level”. According to them, parameters embody the physical and functional aspects of the system and scaling therefore implies an increase or decrease in the variable that quantifies the parameter (the parameter level). The third effect is when a parameter is added or removed from the parameter set (i.e. the ‘parameter set’ is made larger or smaller). In this instance, the system is modified and can be said to be ‘modifiable’.

The above focuses mostly on the intended effects of changes to a system. Unfortunately, however, what usually makes changing a complex system challenging, is

that it could produce numerous undesirable effects. For example, trying to change one component could affect multiple others, and so forth. This is referred to as ‘change propagation’. Several methods exist to predict and understand this change propagation (see for example [46], [47]; [48], [49], and [50]). The aim of developing these methods is ultimately to understand how propagation can be minimised.

2.4.2 Engineering Systems Changeability

In the engineering design and systems engineering literature, the word changeability is often employed as an umbrella term for difficult-to-define change-related ‘ilities’, such as ‘flexibility’, ‘adaptability’, ‘versatility’, ‘robustness’, ‘agility’, and so forth [51], [5], [6], [4], [52]. Changeability is therefore usually considered to be a desirable property of a system, in that a changeable system can be modified (or change itself) at relatively little expense (financial cost, time, or ‘effort’) to meet new, varying, or wide-ranging demands imposed on it.

A ‘Design for Changeability’ (for product systems) approach was proposed by Fricke [53] and subsequently expounded by him, along with Schulz ([44] and [4]). In these works, four aspects of changeability are formulated: ‘flexibility’, ‘agility’, ‘adaptability’, and ‘robustness’. Flexibility is a feature of a system enabling it to “be changed easily and without undesired effects” [44]. Note that, with flexibility, the system is changed from outside its borders (i.e. it does not change itself). This is in contrast to adaptability, which these authors define as a property of a system that allows it to change “itself towards changing environments to deliver its intended functionality” [44]. Robustness was defined as a property of a system enabling it to remain unchanged in the face of “changing environments”, while still delivering its “intended functionality” [44]. Similarly, agility relates to flexibility, as it allows the system to be changed “rapidly”.

Another way in which the word robustness can be used is what some authors refer to as ‘value robustness’ [54], [5]. Value robustness is “the ability of a system to maintain value delivery in spite of changes in contexts or needs” [54]. It can therefore be stated that value robustness is the ‘goal’ of making a system changeable, as such a system will continue to deliver value by being easily changed or changing itself in order to accommodate changes in contexts or needs.

It is important to appreciate that changeability is not necessarily ‘free’. This can be appreciated when considering again the change mechanisms referred to in the previous subsection (also presented by Ross et al. [5]). Each mechanism is a ‘path’ between one state and another and has a certain associated cost – monetary and time expenses to perform the change, or a negative effect on the performance of

the system. Certain paths could therefore be more attractive than others. When a system is designed to be changeable, the design process could also elicit surplus costs, because it could take longer (as additional analysis is required) and consume extra resources. Although it is more expensive, increasing changeability also increases the value of the system. De Neufville and Scholtes [55] state that “flexibility [changeability] provides a two-fold advantage: it limits possible losses and increases possible gains. Both actions increase the overall average value”.

Several attempts have been made to quantify changeability. Many involve the use of the multi-attribute tradespace concept, which is described in Section 2.6.4.3. Ross and Hastings [51] and Ross et al. [5], for example, introduced the measures of ‘outdegree’ and ‘filtered outdegree’. The outdegree of a design is the number of other designs in the tradespace that this design can possibly change into, whereas the filtered outdegree is the number of designs it could change into in an ‘affordable’ manner. These measures have been refined by several authors (see for example the ‘value-weighted filtered outdegree’ [56]), and many further improved metrics for use with tradespaces have been introduced, most notably by Fitzgerald and Ross [14]. Many authors also attempt to measure changeability in terms of how change propagates through the system (e.g. [57]; [58] and [59] – the less severe the propagation, the more changeable the system would be, as there is essentially less ‘resistance’ to change. Most of these metrics are equally applicable to evolvability and are explored in more depth in Sections 2.6.

At this stage, it is important to distinguish changeability from the concepts of ‘product families’ and ‘product platforms’. Pirmoradi, Wang, and Simpson [60, p. 2], quoting Meyer and Lehnerd [61], define a product family as a “set of products that share a number of common components and functions with each product having its unique specifications to meet demands of certain customers”. They continue to state that the product platform is usually considered as the collection of these common parts (quoting Simpson et al. [62]). Together these two concepts have a large field of research devoted to it for which a sizeable literature exists (see [63] and [64] for a comprehensive overview).

Fricke and Schulz [4, p. 346] state that product platforming “focuses mainly on offering a high product variety to customers while reducing development and manufacturing cost” and is “a specific solution approach that addresses a subset of [change-related challenges]”. Conversely, “DfC [Design for Changeability] focuses on incorporating changeability into a system’s architecture in order to enable for foreseen and unforeseen changes within the architecture throughout the systems life cycle, which could include using an existing architecture for possible derivatives” [p. 346]. Platforming and families therefore promote variety, whereas evolvability

relates more to reusing and changing the design over time.

However, changeability could, indeed, be embodied in a product family. The Boeing 737 is an example of this, where each generation (i.e. the 737 originals, the Classics, the Next-Generation, and the MAX) could separately be considered a product family. Changeability (evolvability in this instance), in this case refers to the ease (or difficulty) with which the substantial changes in engines, wings, empennage, and systems *across* the generations were applied over the decades. The product platforming literature has started to be supplemented by research that focus on incorporating changeability in product platforms (see for example Refs [65] and [66]) and generally fall under the term of ‘Flexible Platforms’. The reader is directed to Zhang [67, p. 7] for an overview of this topic.

Finally, many of the concepts relevant to this research (especially commonality) apply to both changeability (including evolvability) and product platforms and families. Authors active in one or both fields are therefore cited throughout the text.

It will now be shown that evolvability is a specific type of changeability. After this, the phenomenon of evolution in aircraft is discussed, followed by techniques for designing for and exploring evolvability during conceptual design.

2.4.3 The Concept of Evolvability

As highlighted by Colombo, Cascini, and de Weck [45], when discussing the changeability of engineering systems, it is helpful to distinguish between the actual physical realisation of the system, i.e. the manufactured product (also referred to by these authors as the ‘instance’) and the abstract representation of the system (also referred to as the ‘design’, ‘architecture’, or ‘holon’). Changeability could be inherent the actual instance, or in the design of the system. ‘Evolvability’ refers to the changeability of the *design* and, more specifically, the degree to which the design can be reused or modified over time. As stated in Chapter 1.1, the following definition from de Weck, Ross, and Rhodes [3] captures this idea well and was adopted for this research:

Evolvability: The ability of a system design to be inherited and changed across generations (over time).

‘Inheritance’, which invokes the biological concept of evolvability, implies that there are ‘traits’ of a design (characteristic to individual parts, subsystems, major components, or even processes) that remain relatively unchanged in subsequent generations of designs. This reuse of design elements in future products suggests

‘commonality’ with the baseline design. The most significant advantage of employing commonality (in the context of engineering design) is a reduction in development cost and risk, as much of the design work and testing does not have to be repeated (see Cameron and Crawley [68] for a general discussion on the benefits of commonality). Referring to the Boeing 737 as an example, the fuselage design has remained mostly unchanged since the original versions (apart from being lengthened). Indeed, the same nose, fuselage cross-section, and tail cone designs were used on all versions, except on the 737MAX, where a new tail-cone design was adopted. Commonality as an enabler for evolvability is discussed in more detail in Section 2.6, along with other relevant enablers.

The word ‘changed’ in the above definition indicates that, though much reuse is desired, modification of a baseline is inevitable for the new design to be competitive. It is common for aircraft manufacturers to ‘re-engine’ existing airframes, or to increase the fuselage length by inserting ‘plugs’ to accommodate more passengers. A less common modification is to change the wing design, which is usually considered a significant redesign, but which could greatly enhance performance. It is important therefore to try to predict what changes may be required in the future and to ensure that the design would be amenable to these.

‘Across generations (over time)’, implies that evolvability is applicable over relatively long periods. In fact, the benefits of evolvability are only reaped at the ‘redesign’ phase of a system’s life cycle [45], which could be many years (even decades) after the original development. The world, however, changes swiftly – the needs of society and the economy fluctuate and, over the past few decades, technology has progressed at an unprecedented pace. An attempt must therefore be made to account for these ‘exogenous’ changes when designing for evolvability.

As can be deduced from the above discussion, evolvability is a particular type of changeability, which concerns the ease at which the *design* can change. More specifically, it can be classified as a subset of the changeability aspect of ‘flexibility’ (as used by Fricke and Schulz [4]), because the change is applied from outside the boundary of the system. The applicable change mechanism is redesign and the change agent is the system design team. Many of the concepts and changeability measures discussed previously therefore apply equally to evolvability. The discussion will turn next to evolvability as a phenomenon in aeroplanes.

2.5 Evolution as a Phenomenon in Aircraft

Whether it was planned or not, many aeroplane designs have undergone significant evolution. This is true for military, civil transport, and business aviation

aeroplanes, as well as some general aviation aircraft. In line with the context of this research, evolution is not considered here in terms of the evolution of generic aircraft technology, design, and manufacturing processes, but of the evolution of individual designs (e.g. the A320 and its family). For an excellent overview of the former (for airliners), the reader is directed to Whitford [69]. A few notable examples of evolution in current and historical aircraft are discussed here, followed by a brief look at the consideration of evolvability in unconventional airliner configurations.

2.5.1 Beginnings

Examples of design reuse and change can be observed from the earliest days of aviation. The Wright Brothers, for example, regularly reused specific features or components on their aeroplanes. For the 1905 Wright Flyer III, the first ‘practical aeroplane’, they reused the actual engine and most of the metal hardware from the Flyer II [70]. They also re-adopted the wing camber of their famous 1903 design [70]. However, the structure was stronger, the tail taller, the control surfaces featured more area, and the aircraft sat higher above the ground than its predecessors [70].

The Douglas Company, perhaps the most successful American aircraft manufacturer before the jet age [71], also significantly exploited evolvability in their designs. The design of the DC-1 (Douglas Commercial 1) was reused and modified to create the larger DC-2 and, ultimately, one of the most important civil aircraft ever – the DC-3. To devise the DC-2, the fuselage of the DC-1 was ‘stretched’ (lengthened by introducing extra frames) [71, p. 162] – a practice that has become widely used. The modifications to the DC-2 to create the DC-3 were much more substantial. The fuselage was widened and stretched; the wings were made stronger and longer; the undercarriage was strengthened; and the tail-surfaces were enlarged. The similarities, however, remain unmissable. All later Douglas propeller transport designs descended from these aircraft and the DC-4, DC-6, and DC-7 were particularly close relatives. The DC-6 was a stretched version of the DC-4 [71, p. 397], whereas the DC-7, in turn, was a stretched version of the DC-6 [p. 469]. In both cases, new technology was incorporated – the DC-6 was the first Douglas transport aircraft that featured a pressurised cabin [71, p. 397]) and the DC-7 featured new materials and more advanced engines [p. 469].

2.5.2 Military Examples

Several case studies of evolution in military aircraft exist in the literature. These include studies involving fighter aircraft, helicopters, and bombers. Some examples

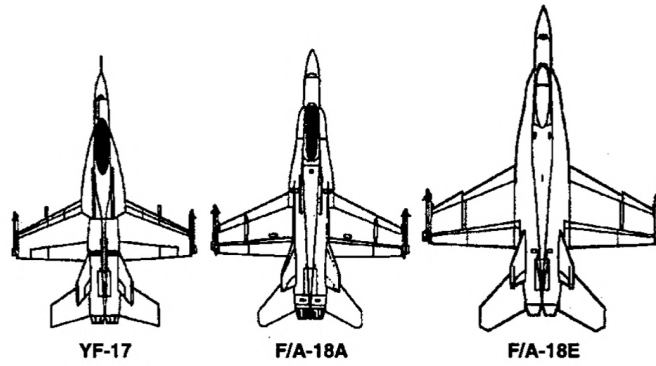


Figure 2.4: Evolution of the F/A-18 (from Refs [72] and [7]).

of fighters and bombers are covered here.

Lim [7, p. 41] state that the evolution of military fighter jets is driven often by the need to improve lethality and survivability. In that work, fighter jet case studies were performed on the Northrop F-5, the Lockheed Martin F-16, and Boeing F/A-18 (see pp. 41-50). With the F-5, there was a substantial increase (almost double) in maximum takeoff weight (MTOW) over the period of about twenty years from the original T-38 design to the F-5E [p. 41]. This was enabled by advances in engine technology, allowing for greater thrust without major increases in engine size and mass [p. 42]. The wing was only enlarged once [p. 42]. In addition, considerable volume was left in the nose so that avionics and weapons equipment could be added to later versions [p. 42].

The F-16, a highly successful aircraft, has been produced in more than 110 versions and has undergone significant evolution [7, p. 43]. Its early evolution is characterised by upgrades in weapon systems, engines, and avionics. Later, its role changed from simply being a lightweight air-to-air fighter to being a ground-attack aircraft with all-weather capability. This necessitated changes to the airframe, which included enlarged wings and empennage surfaces [7, p. 44].

Lim [7] covers the evolution of the F/A-18 in considerable detail. This aircraft has grown significantly in size and capability from the original experimental YF-17 prototype (see Figure 2.4). Of particular note is the redesign of the C/D models to the E/F models in which the fuselage was ‘stretched’, wing photographically scaled, the horizontal surfaces enlarged, new engines provided, amongst others [7, p. 45] (quoting Jane’s All the World’s Aircraft 2006-2007).

Long and Ferguson [73] have also studied the evolution of the F/A-18, this time in conjunction with that of the Boeing B-52 Stratofortress. In their study, the focus is specifically on the provision of ‘excess’ in each of these aircraft. Excess is akin to ‘room-for-growth’ and will be discussed in more depth in Section 2.6. Long

and Ferguson contrast the relative ease (i.e. without undergoing costly changes) at which the B-52 could adapt to new types of missions, with the substantial redesign that the F/A-18 had to undergo. They attribute this to the greater allowance of excess in the B-52. Questions arising from this study include:

- How much excess should be included? The fact that the later versions of the F/A-18 were developed, rather than a brand-new design, suggests that redesign was still the preferable option.
- What types of excess should be provided for? Long and Ferguson [73, p. 9] showed that some types of excess could be ‘fungible’, and finding an appropriate balance would therefore be desirable.
- When is the provision of excess negatively affecting the performance for the original role too severely? It is important to recognise that the original YF-17 was developed as a lightweight low-cost air-superiority fighter. In contrast, the role that the F/A-18E/F fulfils is that of a ‘high-end’ multirole fighter – previously filled by the much heavier Grumman F-14 Tomcat. Again, re-using and adapting the F/A-18C/D design was most likely considered a superior option than designing a new aircraft (or even upgrading the F-14).

These questions apply equally to other types of evolvability ‘enablers’ (discussed in Section 2.6). Answering them is what evolution exploration during conceptual design aims to support and the current research is intended to aid this endeavour, albeit in an indirect manner (see Section 2.6.4).

2.5.3 Evolution in Civil Jet Transport Aircraft

The aforementioned studies on military aircraft evolution are insightful and important. Indeed, unlike transport aircraft, new roles are frequently defined for military aircraft, as novel and unique threats arise constantly. However, as will hopefully become clear in the next chapter, many civil transport aircraft have also, like their military counterparts, undergone significant evolution through many innovative advances, and therefore warrant study. Surprisingly, a comprehensive and cohesive account of the evolution of civil transport aircraft designs could not be found in the literature¹. The author therefore performed a detailed study of design reuse and evolution in the jet transport aircraft of three large aircraft manufacturers – Boeing, McDonnell Douglas, and Airbus. The results of this study constitute a contribution to knowledge in itself and are presented separately in

¹A comprehensive study covering the changes of commercial aircraft architectures from the Douglas DC-3 to the Boeing 787 have recently been published by Kellari, Crawley, and Cameron [74]. However, this study focuses on architectural changes as a collective and not re-use and commonality across specific designs.

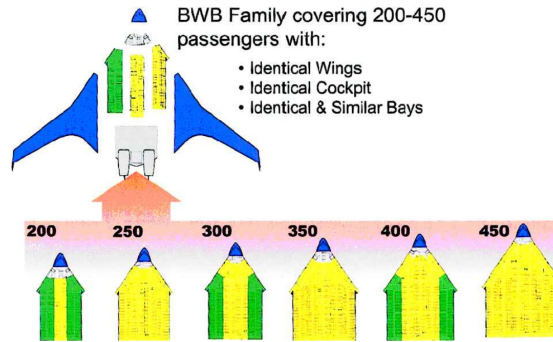


Figure 2.5: Commonality in a Blended-Wing-Body family (Liebeck [75]).

the next chapter. Methods to support design for evolvability specifically for civil airliners have been proposed, however, and are explored in Section 2.6.

2.5.4 Unconventional Configurations

Of particular interest in this work is evolvability across aircraft designs that have different configurations. Not much could be found on this topic in the literature, but this phenomenon does, however, occur in reality, as can be seen from design reuse in the early Boeing and Douglas jetliners (see Chapter 3 for a more in-depth discussion on this). Some descriptions of families of unconventional aircraft have indeed been found, albeit that all members share the same configuration. One such example is from Liebeck [75] and concerns the description of a family of blended-wing-body passenger aircraft (see Figure 2.5). In this description, Liebeck demonstrates how the wing and cockpit design could be retained across different family members, while inserting bays to increase payload capacity. This would have the effect of automatically increasing span, which is beneficial in terms of countering the additional mass. Another example can be found in Scholz [76], who demonstrated how a box-wing design could be stretched to create additional family members (see Figure 2.6).

These studies are essential, and the author is of the opinion that growth versions should be studied for every novel configuration proposed. However, for manufacturers to adopt new configurations, they would likely prefer to offset the major risks involved (associated with novel configurations) by gradually evolving existing conventional designs to these. Again, it is hoped that this research establishes the tools and framework to facilitate such studies.

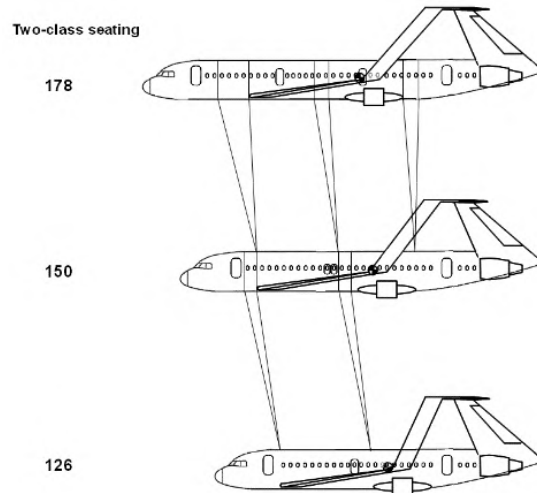


Figure 2.6: A single-aisle box-wing aircraft family (from Scholz [76]).

2.5.5 ‘Variant’, ‘Version’, ‘Derivative’, and ‘New’ Aircraft

To prevent confusion, a discussion on the terms ‘variant’, ‘version’, ‘derivative’, and ‘new’ design is needed. Zhuravlev and Zhuravlev [77, p. 173] state that a ‘new’ aircraft is one in which there is a:

- new wing
- new fuselage
- new number and/or position of engines

A new wing, according to them, is one in which the following is different from the baseline aircraft:

- set of aerofoils
- type of wing box structure
- main geometric parameters:
 - span (only in some cases)
 - sweep angles of leading edge and quarter-of-wing-chord line
 - length of aerofoil chords, taking into account their distribution along the span
 - wing box dimensions, including shape and dimensions of cross sections of its elements.
- structural materials of the main load-bearing elements
- types of connections of wing box elements

Finally, a new fuselage has (a) different:

- cross-section shape

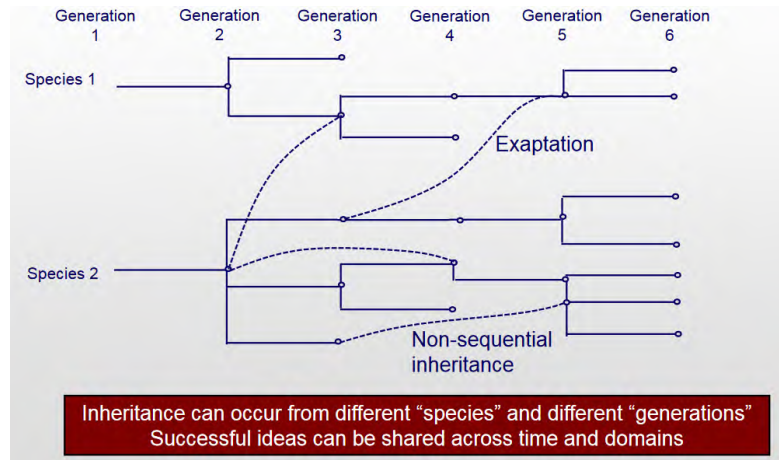


Figure 2.7: Technological inheritance (from Beesemeyer et al. [79]).

- type of structure
- main geometric parameters
 - diameter
 - number of passenger cabin decks and their positions
 - geometry and dimensions of nose and tail parts
- structural materials of the main load-bearing elements
- types of connections of elements

All other changes to the main components and onboard systems of an existing aircraft result in a version of that aircraft [77, p. 173]. ‘Version’ and ‘variant’ are synonyms and often mean the same as ‘derivative’, but a derivative refers specifically to an aircraft design that is based on that of another, already existing aircraft.

These terms apply to a closely related family of aircraft. The research in this thesis is more generic, however, because it could be desirable to capture the benefits of *any* design re-use or ease of change when a specific design is used as baseline. The word ‘descendant’ is therefore employed here to refer to a subsequent design to a baseline that has non-zero commonality with that baseline. Thus, no distinction is made between variant, version, or derivative, and these are used in this research as a subset of ‘descendants’. This is more in line with the word ‘evolution’, where traits are retained over generations, even though the specimens in each generation may be classified as different (new) species. The reason for using the term ‘descendant’ is exactly that – it is envisioned (for future work) that the results of this research be used to identify and explore evolvability over the *entire* product portfolio of a manufacturer, and not just within one family. Such ‘cross-species’ transfer of design elements is akin to ‘exaptation’ (see Figure 2.7) and is common in technology, as well as useful (see Beesemeyer [78, p. 84]).

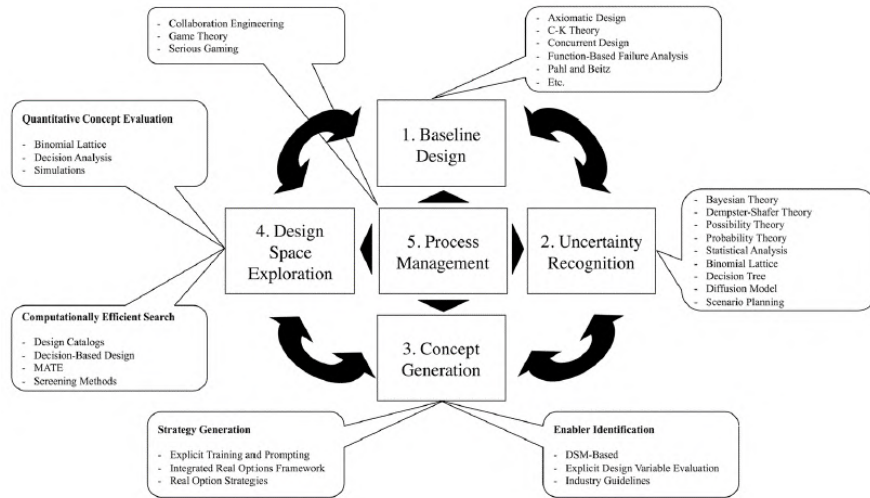


Figure 2.8: Taxonomy of procedures to support the design of engineering systems for flexibility (from Cardin [80]).

2.6 Design for Evolvability

In this section, an overview of designing for evolvability is provided. The purpose is to introduce the reader to the many approaches that have been proposed. Note that, in line with the discussion on evolvability and changeability in Section 2.4, many methods introduced here are more widely applicable than to just evolvability.

Cardin [80] proposed a comprehensive taxonomy for procedures that support the design of engineering systems for flexibility (see Figure 2.8). This taxonomy is directly applicable to evolvability and used in this section to guide the discussion. As can be seen in the figure, the taxonomy denotes five phases of the design for flexibility (evolvability) process. Several procedures that Cardin identified to support each phase are also shown. A subsection is devoted to each of the phases. Only the procedures of each phase that are most relevant to the current research are discussed. For explanations of the other procedures, the reader is directed to the study of Cardin.

2.6.1 Baseline Design

According to Cardin [80], “Design for flexibility must start from an existing configuration”. This phase therefore encompasses the creation of designs as is normally done. In aircraft conceptual design, this will involve the use of existing knowledge, current (or ‘near-future’) requirements, concept generation techniques, sizing, and simulation methods to create many thousands of design options (of different sizes but possibly also of different configurations).

Lim [7] prescribes a slightly different order of activities with a method specifically applicable to aircraft evolvability, called ‘Evaluation of Lifelong Vehicle Evolution’ (EvoLVE). In EvoLVE, the current and future needs are established before the baseline designs are generated [7, pp. 89-96]. After the requirements have been determined, a baseline platform(s) is down-selected which has “the greatest potential to meet the current and future requirements with the least cost for further evaluation” [p. 97].

Although EvoLVE could conceivably be applied to multiple baseline platforms, the demonstrations in Lim’s work are for a single baseline platform (i.e. configuration) only. As will be explained in Section 2.8, it may be exceedingly beneficial to keep the design space open for longer, and this should include retaining different configurations further into the design process. Subsequently, as already mentioned, the emphasis in this research was on providing techniques to support the ability to explore multiple configurations. The work herein therefore complements the EvoLVE method and makes it more powerful.

2.6.2 Uncertainty Recognition

This is an important phase in which it is attempted to capture the effects of the changing physical, political, social, economic, industrial, and technical environment/s on the system over time. For civil aircraft, another increasingly important uncertainty driver is that concerning environmental regulations. The aim in this phase is both to identify and to model the major sources of uncertainty that could affect the value of the system over time [80, p. 5]. The field of uncertainty quantification and management (UQM) is vast and falls beyond the scope of this research. However, note that any of the techniques/theories shown for the uncertainty recognition phase in Figure 2.8 could be used in unison with the methods proposed in this thesis.

It is especially important in this phase to account for how the changing world will translate into changing requirements. Establishing what the requirements could be is especially challenging, given the time-scales and number of stakeholders involved. Expert elicitation is often employed and a method appropriate to the challenges in aerospace is the Delphi-based Systems Architecting Framework (see Aliakbargolkar [81] and Aliakbargolkar and Crawley [82]).

It is also crucial here to identify what technologies may (or may not) become available in the future. Aircraft manufacturers therefore often invest heavily in studying emerging technologies and how these could be exploited. Traditionally, the maturity of a technology is established by assigning a ‘technology readiness

level' (TRL) to it. More quantitative and systematic approaches that incorporate the vehicle in the process have been proposed recently. These include a method to link uncertainty reduction to the progression of TRLs by Schwartz and Mavris [83] and a concurrent engineering approach for model-based technology roadmapping by Knoll, Aliakbargolkar, and de Weck [84]. Approaches to determine the effects of infusing new technologies into existing architectures have also been proposed (see for example de Weck [85], Smaling and de Weck [86] and Suh et al. [87]).

Knoll, Aliakbargolkar, and de Weck [84] distinguish technology assessment (determining the current state of the art for a specific technology), technology roadmapping (identifying possible technology evolution paths), and technology planning (selecting a specific path/s), as three separate consecutive phases in the Technology Planning and Roadmapping (TPR) process. These activities often take place before and separately from any specific aircraft programme and the results for the technology planning phase would be used in an evolvability study to determine which designs would be more amenable to having certain technologies incorporated. However, the author believes that the techniques proposed in this research could greatly assist the TPR process in that they may help identify which designs/configurations are compatible with which technologies and vice versa.

2.6.3 Concept Generation

'Concept generation' refers here to devising concepts that could be incorporated into the baseline designs to facilitate addressing the uncertainty-related factors identified in the previous step. Each concept has an associated 'strategy' and 'enabler' [80]. Strategy refers to how the system would be changed when uncertainty resolves, whereas enablers refer to the specific features in the design and management practices that would facilitate the particular changes [80, p. 5].

2.6.3.1 Strategy Generation

The generation of strategies are beyond the scope of this research and only a short overview is provided here. The techniques proposed in this research could be used with any of the strategies discussed below. Real option valuation is often used to identify preferred strategies (see for example Refs [55], [88], [89], [90] and [91]). A real option is the right, but not the obligation to some future action [55, p. 265]. An option in this case refers to providing some feature in the design that would ease change at a later point in time, if necessary.

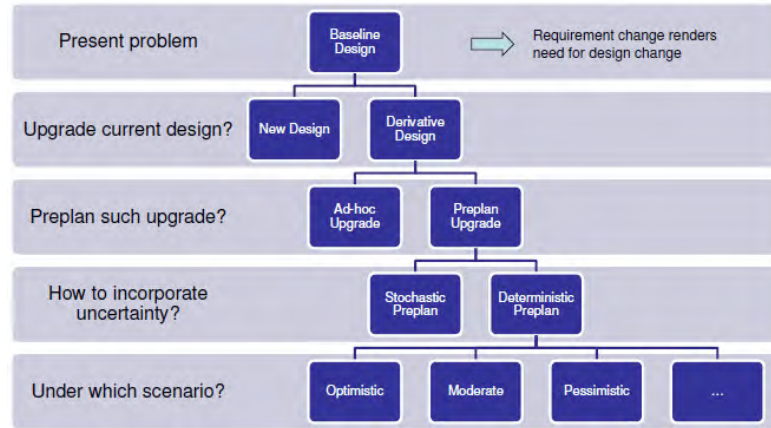


Figure 2.9: Taxonomy of evolution paths (from Lim [7, p. 100]).

Standard evolution strategies specific to military acquisition have also been formulated. These include ‘Pre-Planned Product Improvement’ (P3I) [92] and ‘Evolutionary Acquisition’ (EA) [93] in conjunction with ‘Spiral Development’ (SD) (see Lim [7, pp. 15-22] for an overview). Lim [7, pp. 99-107] also presented a taxonomy of managerial strategies appropriate for evolution in aircraft design (see Figure 2.9). These strategies could be modelled and applied in the exploration process in order to gauge their effectiveness.

2.6.3.2 Enabler Identification

Naturally, much research has been devoted to identifying enablers in flexible design. The literature on this topic is vast and there is much overlap. The classification in Figure 2.8 for enabler identification is therefore used again as a guide. The classification distinguishes between ‘design structure matrix-based methods’, ‘industry guidelines’, and ‘explicit design variable evaluation’. Each are discussed briefly below. This is followed in Section 2.6.3.3 by a discussion on how these enablers relate to commonality and how commonality is used in this research.

Industry Guidelines

Industry guidelines are lessons and principles which could facilitate flexibility in design [80]. Fricke and Schulz [4] have provided a number of these principles to promote changeability in general. For the current discussion the most relevant are:

- *Ideality/Simplicity*: Ideality concerns the reduction of complexity and relates to the information axiom of Axiomatic Design (see Suh [94]). If possible, the system should be made simpler by minimising the number of interfaces and secondary functions, as well as reducing the use of and focusing on using existing resources in the system, supersystem or environment [4, p. 358].

Tilstra et al. [95], propose a similar principle, called the ‘parts reduction principle’, in which it is endeavoured to share closely-related functions in modules or parts and using duplicate parts as far as possible. An example of the ideality principle in aeroplanes is the practice of using fuel as a cooling fluid instead of incorporating a dedicated liquid cooling system. This reduces the number of parts and hence the complexity in the system, which makes it easier to change.

- *Independence*: This principle involves ensuring/attempting that each system function is satisfied by a separate design parameter (or subsystem, component, or part) and originates from the first axiom of Axiomatic Design [94]. It ensures that change propagation remains limited. An example from civil aircraft is the practice of hanging engines in pods under the wing, instead of being embedded inside the wing. If they were to be embedded in the wing, and it is desired to incorporate new engines, the wing might need to be modified significantly to accommodate the new engines. Another example is the use of the tube and wing configuration where the function of providing lift is separately fulfilled by the wings and the function of carrying payload is fulfilled by the fuselage. These components can therefore be modified independently.
- *Modularity/Encapsulation*: Modularity, being perhaps the most intuitive of the enabling concepts, is studied widely and often cited as a principle to promote evolvability (see for example Refs [95], [96], and [78]). In a modular systems architecture, the functions are clustered into several physical modules, which are loosely coupled with each other [4]. This promotes the reuse of components, modules, and, ideally, large sections of the architecture [4]. Modularity is therefore closely related to commonality. According to Pirmoradi et al. [60, p. 5], “modularity decomposes functions into independent groups, while commonality clusters the components and functions based on similarity or other criteria”. Subsequently, modularity supports commonality (along with the other principles listed here), but also makes it easier to change individual components/modules. Commonality is discussed in more detail further on. Modularity also relates strongly to and supports independence, and vice versa [4]. The example of the tube and wings configuration therefore applies again, where the fuselage (tube) and wing act as separate independent modules.
- *Scalability*: As the name suggests, scalability refers to the ability to increase/-decrease specific design parameters, or the number of multiple identical elements [4]. The design parameters could be geometric, such as length, area, and volume, but also non-geometric, such as thrust and power. Perhaps

the most widely used example of scalability in civil transport aircraft is the lengthening (stretching) or shortening (shrinking) of the fuselage to increase or decrease payload capacity. The tubular construction of modern transport fuselages is particularly amenable to this practice, whereas the length of a more aerodynamically optimised, non-tubular fuselage would require considerable more redesign work. As will be discussed in the next chapter, the Boeing 737 was perhaps easier to stretch than the rival McDonnell Douglas DC-9, because it had a short ‘fat’ fuselage (seating six passengers abreast), which required less lengthening to increase capacity, compared with the five-abreast fuselage of the DC-9.

- *Redundancy*: Redundancy is an essential enabler for evolvability and supports scalability extensively. It may refer to the duplication of certain components within a system architecture, but, for evolvability, it is more often considered as ‘excess’, ‘reserves’, or room (margin) for growth. These terms are used interchangeably throughout the text. Excess is defined by Allen, Mattson, and Ferguson [97, p.] as “a resource, embodied by a system, which is not committed to any of the system’s initial set of customer requirements.” There are several types of excess (see [97, p. 2]), but for the current discussion the most relevant are excess in geometrical parameters, i.e. length, area, and volume (space) and structural strength. Several authors have quantified excess (see for example [97], [98]) and shown that it is fungible (i.e. different types could be traded off with each other) [73]. A framework that could be used to trade off different types of excess is from Guenov et al. [33]. In this paper, the word ‘margin’ is employed, but the approach would apply equally to excess. Examples in aircraft design are plentiful and excess is perhaps the most widely used enabler in the aircraft manufacturing industry. For example, Raymer [9, p. 853] points out that the undercarriage of new aircraft is often designed with an increase of 25% in gross weight in mind. Zhuravlev and Zhuravlev [77] have also written specifically about reserves to facilitate creating derivative passenger aircraft. They state that reserves in wing area and undercarriage length are particularly ‘reasonable’, because it is exceedingly difficult to change these [p. 175]. Longer undercarriage legs, in particular, provide reserves for the rotation angle at take-off, which allows the fuselage to be made longer (see Figure 2.10). It also allows for larger engines to be fitted to the design for subsequent variants. Conversely, Zhuravlev and Zhuravlev state that reserves in fuselage volume, fuel tank capacity, structural strength, and engine thrust are not necessary as these could be changed easily [p. 175]. According to them, it is particularly ‘easy’ to implement local reinforcement to increase structural strength and volume

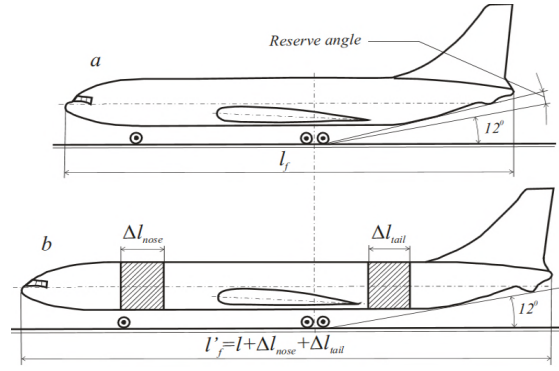


Figure 2.10: Reserves in undercarriage legs allow lengthening of the fuselage (from Zhuravlev and Zhuravlev [77]).

is usually available to incorporate additional fuel tanks. The findings of the case studies presented in the next chapter seem to support these claims. Finally, designers should remain aware that too much excess could affect the performance of the baseline such that it may be unsuccessful in service, which would inevitably cause the whole family to fail.

For evolvability, Beesemeyer [78, p. 164] adds to the above the principles of ‘Leverage Ancestry’ (incorporating successful design choices from previous generations); ‘Mimicry’ (incorporating successful design choices from other systems or domains); ‘Disruptive Architectural Overhaul’ (modifying large parts of the architecture at a time); and ‘Resourceful Exaptation’ (reusing design elements from another system to perform a different function in the new system). Leverage Ancestry and Disruptive Architectural Overhaul seem to be more relevant at the evolvability exploitation stage (i.e. during the design of the descendant), but could be appropriate when designing the descendant to be more evolvable for future benefit.

Design Structure Matrix-Based Methods

Design structure matrix-based methods exploit the power of design structure matrices (DSMs) to identify where in a system design flexibility should be embedded [80]. According to Eppinger and Browning [99, p. 2], a DSM is “a network modelling tool used to represent the elements comprising a system and their interactions, thereby highlighting the system’s architecture (or designed structure)”. It is a $N \times N$ square matrix in which the interactions between the N elements of the system are mapped to each other [99, p. 2]. For design for flexibility purposes, DSMs are often used in the analysis of change propagation within a system (see for example Refs [65], [100], [59]) or to study the relationship between product architecture and evolvability (e.g. Luo [101]). In these analyses, the components identified as ‘multipliers’ (i.e., components that cause more change than they absorb when affected by a change event [65]) are usually the prime candidates in which to embed

flexibility (i.e make them less likely to change by, for example, over-sizing them or rendering them ‘adjustable’). As will be shown in Chapter 4, DSMs are used in this research to determine when components across two airframes could be considered similar enough to undergo further, more detailed similarity assessment.

Explicit Design Variable Evaluation

Explicit design variable evaluation involves singling out the most important design variables and qualitatively investigating the effects they may have on the life cycle performance of the system, if changed [80, pp. 6-7].

2.6.3.3 Discussion: Enabler Identification and Commonality

The above procedures attempt to promote design reuse and facilitate change where necessary for descendant products. As already stated, design reuse constitutes commonality, which is an important concept in this thesis. This section is devoted to defining commonality and to illustrate how it is applied in the current research.

According to Boas [102], commonality is the “sharing of components, processes, technologies, interfaces and/or infrastructure across a product family”. It provides many life cycle benefits [68], but the focus here is on its potential to reduce development time and cost (often referred to as Research, Development, Test, and Evaluation cost or RDT&E [9, pp. 705-707]), manufacturing cost and time, and certification effort. These are elaborated upon below:

- *Reduction in development time and cost:* Development time and cost could be reduced, because the total ‘development scope’ is reduced [102, p. 41]. However, this reduction could be offset if the common modules are more complex than the unique ones that they replace [102, p. 41]. Furthermore, development of the first product is generally expected to cost more than if no commonality was planned, whereas subsequent variants are expected to cost less [102, p. 41]. Another significant reason for lower derivative development cost and time is that expensive tooling (developed for the baseline) is already available.
- *Reduction in manufacturing time and cost:* Manufacturing cost is expected to be lower for the first unit of a subsequent variant, as it would be positioned further along the production ‘learning curve’ of a component that is already being manufactured for another product [102, pp. 42-45].
- *Certification effort:* A major advantage of commonality in aircraft families relates to so-called “grandfather rights”. These are ‘asserted’ when derivatives are certified with only amendments to the original type-certificate of the baseline, which significantly reduces the certification-related work required

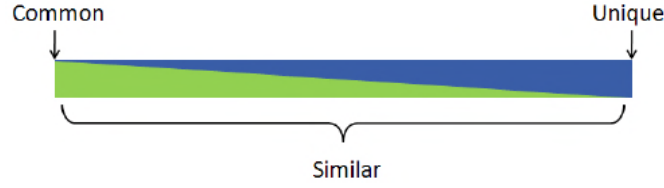


Figure 2.11: The commonality spectrum (from Boas [102, p. 21])

for the new member.

It is important to note that commonality usually manifests on a spectrum [102, p. 21]. A pair of components/parts/subsystems can be anywhere on this spectrum (see Figure 2.11), which stretches between ‘common’ and ‘unique’, with a degree of ‘similarity’ (i.e. a ‘cousin’) anywhere between.

Commonality is often measured by using either a commonality metric, or estimating the RDT&E and manufacturing costs reduction benefits. A commonality metric (also referred to as a commonality score or index) is a quantitative measure of the commonality across product variants that aids in product platforming- (and flexibility-/evolvability-) related decision-making [103]. Many such metrics have been proposed and, for an overview, the reader is directed to Pirmoradi et al. [60, pp. 12-16]. An example of a simple metric is as follows (from Johnson and Kirchain [103, p. 479]):

$$C_{\phi} = \frac{\sum_{i=1}^d \left(\phi_i \frac{\sum_{j=1}^m \gamma_{ij} - 1}{m-1} \right)}{\sum_{i=1}^d \phi_i} \quad (2.1)$$

In this equation, C_{ϕ} is called a ϕ -weighted metric and could be between 0 (no-commonality) and 1 (full commonality); ϕ is a weighting parameter, such as mass or cost; m is the number of products variants (in the product family); d is the number of parts considered; and γ_{ij} is a binary operator that is equal to one if variant j contains component i , and zero otherwise [103, p. 478].

For cost estimation, a simple normalised cost model, which takes into account the benefits of commonality in aircraft design and development was proposed by Markish [104]. Although this cost model is useful for ordinal comparisons, it has a number of limitations that should be taken into account when using it. This is because it is based on development data for the Boeing 777 and other publicly available cost estimation methods and is therefore likely not generally applicable. It also likely does not even provide a correct estimate of the actual 777 development cost (as admitted by Markish [p. 58]). However, it should be noted that, for

airliners, true development/manufacturing costs and cost-reduction benefits due to commonality are usually not publicly available. In addition, cost-estimation is an extremely complex activity, which is outside the scope of this research.

In the enabler identification methods described above, the focus is usually on investigating a single system architecture design. However, as was discussed in the motivation section of this thesis (see Chapter 1.2), to support evolvability exploration, there is a need for methods that could predict similarity (i.e. commonality) *across* architectures/designs. Note that the commonality predicted could be a result of ‘luck’ or from successful application of the above enablers. In other words, the prediction method would be independent of the enabler generation and would act as an evaluation of the commonality achieved.

2.6.4 Evolvability Exploration

In the design space exploration phase, here termed ‘evolvability exploration’, the design space is searched for the most promising design options. Promising designs in this case would be those that perform well in the near-future timeframe *and* could be changed with relatively little redesign effort to other designs that provide value in ‘far-future’ timeframes (i.e. subsequent generations). The effects of possible ‘decisions routes’ are also investigated here. According to Cardin [80, p. 7], this phase involves using “quantitative procedures to evaluate concepts, and computational procedures to speed up the analysis”. These are discussed separately below. The result is (ideally) a set of recommended concepts that could provide better life cycle value than the baseline designs (resulting from the procedures in Section 2.6.1).

2.6.4.1 Quantitative Concept Evaluation

Quantitative concept evaluation involves calculating some metric, for each of the design options and decision paths (created in the strategy generation phase) considered, to evaluate the flexibility they may provide. The metric is often net-present value (NPV) if the necessary information is available to calculate this. Other means involve tracking utility (see Section 2.6.4.3), or key performance parameters and cost over multiple timeframe scenarios, such as in Lim [7]. Sometimes, a dedicated evolvability metric is defined, which could be used as an objective function in an optimisation problem. An example of such a metric is the evolvability measure, based on ‘usable excess’, by Watson et al. [105].

Cardin, mentions three procedures to perform quantitative concept evaluation: decision analyses, binomial lattices, and simulations [80, p. 7]. All these could be

used to analyse the combination of the concepts themselves, as well as the effects of different decision-making paths. The study of decision-making paths falls outside the scope of this research and, for examples, the reader is referred to Silver and de Weck [106] (time-expanded decision networks) and Babajide, de Neufville, and Cardin [107] (decision trees). For the current work, performing simulations constitute the most relevant method, as it fits with existing practice in aircraft conceptual design. With simulations, the designs and the effects of uncertainty (and decision paths) are modelled explicitly and, using these, the metrics are computed.

2.6.4.2 Computationally Efficient Search

When flexibility and evolvability are considered, the design space is often prohibitively large and strategies or methods are required to aid in efficiently and systematically searching it for promising solutions [80, p. 8]. Cardin mentions decision-based design, screening methods, design catalogues, and multi-attribute tradespace exploration (MATE) as appropriate procedures for this phase.

Decision-based design involves methods that employ a combination of decision-theory and determining the value that stakeholders assign to different designs (often measured as ‘utility’) to search for promising designs (see for example Refs [108] and [109], as quoted by Cardin [80]).

‘Design catalogues’ could be used when it is not desirable or possible to reduce model fidelity. In this method, the multitude of scenarios that could unfold are reduced to a much smaller set of representative scenarios. Flexible designs are then matched to this set to create a ‘catalogue of flexible operating plans’. This limits the number of optimisation and simulation runs needed [80, p. 8].

Screening methods involve using simplified models, optimisation, and design-of-experiments techniques to obtain a shortlist of candidates for further investigation [80, p. 8]. The EvoLVE framework of Lim [7] and Lim and Mavris [110] (see Figure 2.12) provides particularly notable examples of these types of methods and demonstrates them with an application to aircraft. In EvoLVE, Steps 7 to 9 constitute three different options for exploring the design space [7, pp. 112-116]. These could be followed in sequence, or performed separately, as desired. The design space in this case is a ‘two-stage’ design space – the first stage is for the near-future requirements, whereas the second stage is for the far-future requirements. For Steps 7 and 8, two-stage optimisation problems, as formulated in [7, pp. 62-86], are respectively solved deterministically or stochastically to minimise acquisition cost in both stages. Step 9 describes an interactive decision support framework that uses two-stage contour plots (for constraint analysis) and multivariate profilers [7, p. 118]. The power of this interactive framework is that

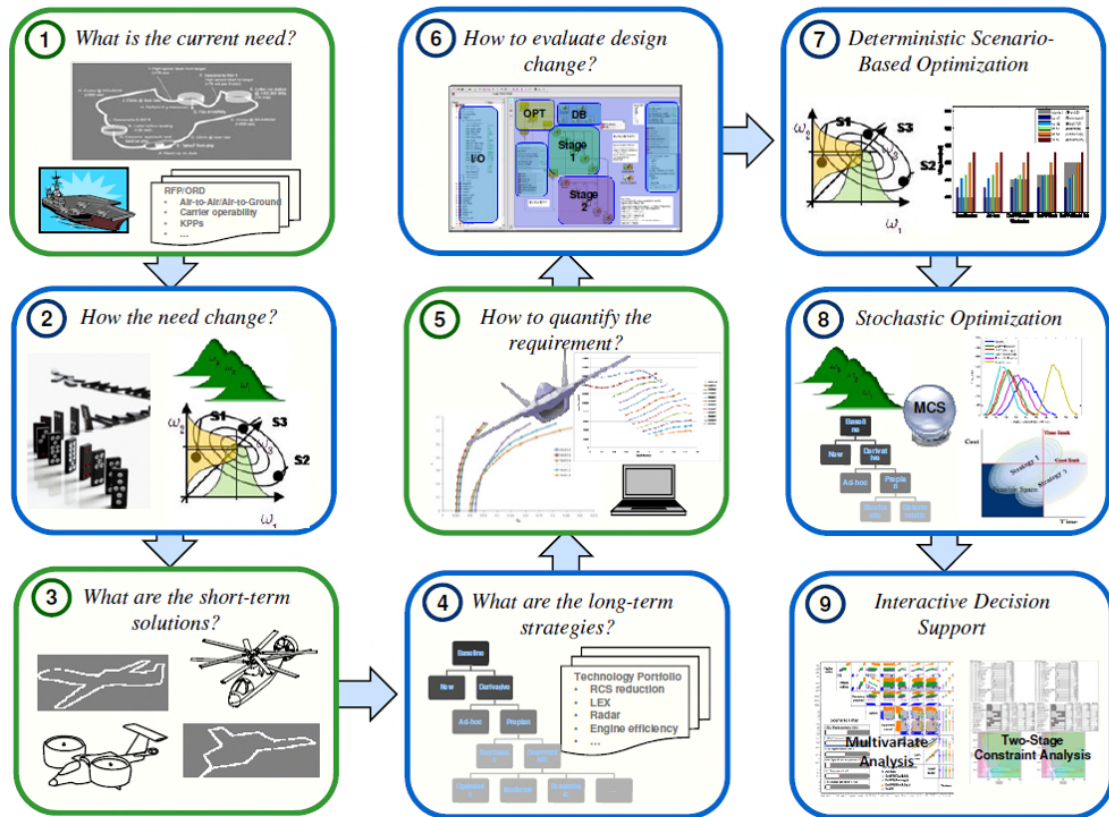


Figure 2.12: Overview of the EvoLVE method (from Lim [7, p. 89]).

it enables the designer to interactively vary requirements and design variables for both stages and immediately see the results.

Although any of the above procedures could potentially be used in conjunction with the results of this thesis, MATE was chosen specifically to demonstrate the techniques proposed. This was because of its simplicity, concise manner of representing information, and widespread application to changeability studies. It is described in more detail in the next subsection.

2.6.4.3 Multi-Attribute Tradespace Exploration

Multi-attribute tradespace exploration (MATE), also called MATE-CON (for MATE with concurrent design) is an approach in which formal decision theory (particularly multi-attribute utility theory) is incorporated into model- and simulation-based design [111]. As will be shown here, it has proven to be a powerful approach to explore changeability in complex system design and numerous methods have specifically been developed that exploit it to support this activity. This was the primary reason it was chosen as the exploration paradigm for the current research. This section is devoted to the definition of MATE and associated methods, a short overview of the main changeability and evolvability techniques developed in unison

with MATE, as well as a description of how it is employed in the current research. According to Ross, Hastings, and Warmkessel, a tradespace [111, p. 21] is the “space spanned by completely enumerated design variables.” It is not the design space, as such, but the potential solution space [111, p. 21] in which designs could be traded off with each other.

In MATE, the preferences of all the stakeholders are encapsulated in a single metric – utility. This ensures that the experts and decision makers are driven by a common goal, which is to find affordable designs with high utility [111, p. 21] (hence the addition of concurrent design). ‘Utility’ is a dimensionless parameter that is used as a measure of the “perceived value under uncertainty” of a specific attribute (or sets of attributes) of interest [112, Chapters 1 and 4]. In the context of complex systems design, these attributes are associated with the design options under consideration. Examples of attributes in transport aircraft could be, for example, reduction in fuel-burn and noise emissions, turnaround time, or something more difficult to quantify, such as passenger comfort. In utility theory, the preferences for an attribute of the designs under consideration are mapped to a “normalised value-under-uncertainty function”, i.e. a ‘utility function’ [111, p. 22]. In ‘multi-attribute utility theory’ (MAUT), the single-attribute utility functions are combined into a single function [112, Chapters 5 and 6], [111, p. 22]. There are assumptions that must be kept in mind to correctly set up a multi-attribute utility function. These are discussed in [113]. If the so-called ‘preferential’, ‘utility’, *and* ‘additive’ independence assumptions hold [113, pp. 178-179], then the utility function can take the form of a simple weighted sum, as follows:

$$u(\mathbf{x}) = \sum k_i u_i(x_i) \quad (2.2)$$

where

$$\sum k_i = 1 \quad (2.3)$$

In equation 2.2, $u(\mathbf{x})$ is the multi-attribute function, $u_i(x_i)$ is the single-attribute utility function associated with attribute x_i , which returns a value between 0 and 1, and k_i is a weighting factor reflecting the importance of attribute x_i . As can be seen, the value returned by $u(\mathbf{x})$ would be between 0 (minimum utility) and 1 (maximum utility).

A multi-attribute tradespace (MAT) is a two-dimensional tradespace in which the ordinate is typically multi-attribute utility and the abscissa the cost of the system. The cost could be total life cycle cost, operating cost, or development and manufacturing cost. This plot is populated by all the enumerated designs such that each design is represented by a utility/cost point (see Figure 2.13). The

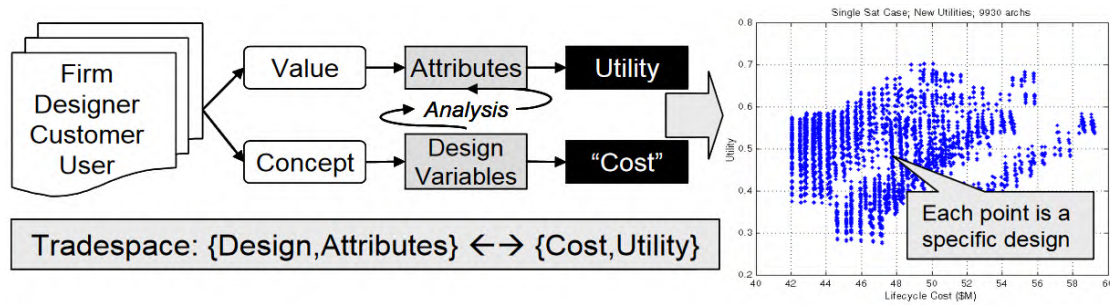


Figure 2.13: Multi-attribute tradespace definition
(from Ross and Hastings [51]).

MAT therefore enables the decision-makers to view the value and cost of all the designs under consideration in a single plot, regardless of the different systems architectures and configurations the designs may embody. Many techniques have been developed to enhance the exploration of complex system designs using the MATE paradigm (see for example Refs [114], [115], and [116]).

Using the MAT, Ross [117] and Ross and Hastings [51] devised the concept of ‘filtered outdegree’. The ‘outdegree’ of a design is the number of other designs in the tradespace that this design can change into [51]. There is a cost (expenditure of resources) associated with such a change and there may be multiple ways in which one design could change into another. Therefore, the ‘filtered outdegree’ was devised to count the number of designs the selected design could change into *affordably* [51]. Filtered outdegree is thus a simple measure for the changeability of a design.

The temporal nature of the environment in which complex systems operate has also been accounted for by combining MATE with a process called ‘Epoch-Era Analysis’ (EEA) [118], [117]. In EEA, the full life cycle is referred to as the ‘era’, whereas epochs refer to periods of time within that life cycle in which the context (factors exogenous to the system) remain ‘constant’ and the system provides fixed value [118, p. 5]. Each epoch is subsequently characterised by “static constraints, available design concepts, available technology, and articulated attributes” [118, p. 5]. Each epoch can therefore be represented as a separate MAT. As a new epoch dawns, the changes in requirements, available technologies, regulatory environments, and so forth, could have an effect on the value of the system and its utility may increase or decrease. Figure 2.14 illustrates this graphically – in Epochs 1 through 4, the example system does not change, as it meets or exceeds expectations. However, in Epoch 5 – if left unchanged – the system falls below expectation, whereas – if changed appropriately – it again exceeds expectations. EEA (also referred to as Dynamic MATE analysis) is a computational scenario

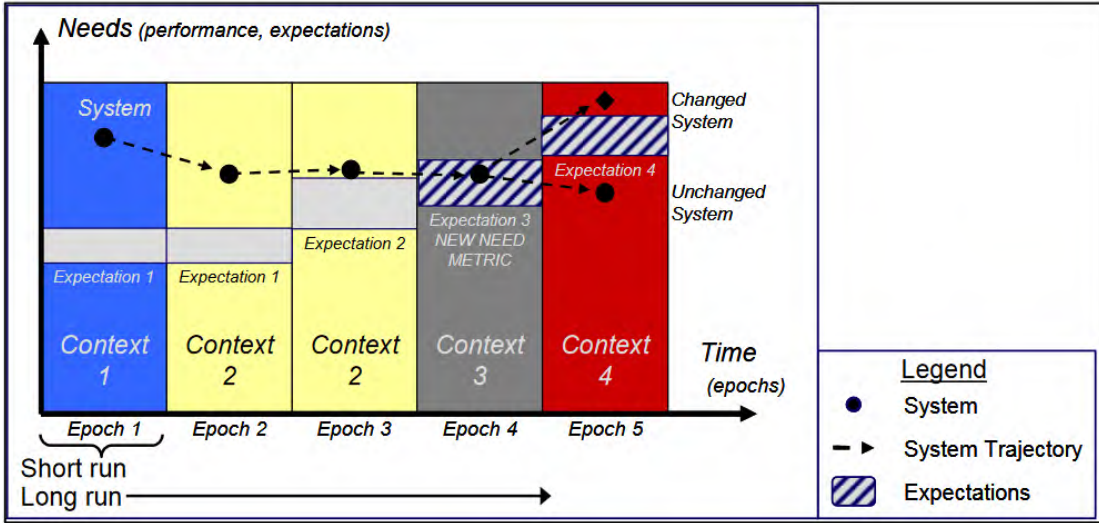


Figure 2.14: System performance and expectations over time represented as epochs (from Ross and Rhodes [118, p. 6]).

planning technique that employs this concept [117], [119]. Multiple eras could be created, with different combinations of epochs each, which would constitute the different possible life cycle scenarios that the system may face.

Techniques were subsequently introduced to expand the use of EEA in performing scenario planning (e.g. Refs [119], [120], and [121]); to investigate several types ofilities for complex systems [122]; and to create further metrics for changeability, such as the ‘Value Weighted Filtered Outdegree’ (VWFO) [56]. Like the filtered outdegree, the VWFO measures how many designs a selected design could change into, but also captures the difference in utility values from the originating designs to the affordable destination designs [56, p. 6].

Improved metrics were introduced by Fitzgerald and Ross [14]. These include the ‘Effective (fuzzy) Normalized Pareto Trace’ (eNPT/efNPT) and ‘Fuzzy Pareto Shift’ (FPS), and ‘Available Rank Improvement’ (ARI). The eNPT metric for a given design captures the percentage of epochs in which the design is on the utility-cost Pareto front, whereas efNPT measures the percentage of epochs it is within the ‘fuzzy’ Pareto front (i.e. within a specified distance of the frontier). These two measures therefore capture robustness, i.e. how the design is performing over time without changing. FPS, on the other hand, measures the difference in ‘Fuzzy Pareto Number’ (FPN) over an epoch change in which the system has undergone a change itself. FPN is a measure of how close the design is to the Pareto front in comparison with all the other designs in the tradespace [14]. The value of FPN for a design could be between 0 (on the Pareto front) and 100 (the furthest point from the Pareto front) [14, p. 3]. FPS is therefore a measure of how changeable/evolvable

the system is. The ARI metric for a design determines how much the utility of this design could increase using a specific change mechanism.

MATE and EEA have also been applied specifically to the exploration of evolvability in complex subsystems. For example, Derleth [123] demonstrated how MATE could be used in Evolutionary Acquisition. Fulcoly et al. [121] and Fulcoly [120] introduced design for evolvability (DfE) as an additional design variable, which was modelled as an increase in mass, but with an associated development time and cost reduction. Commonality and ease of change are therefore assumed to be available already for exploitation.

MATE, EEA, along with the associated metrics and exploration methods are powerful. However, there are still some limitations and unexploited opportunities remain. These limitations and opportunities apply equally to all the other exploration approaches discussed before and will be addressed shortly (see Section 2.6.6). However, before proceeding to that discussion, a brief introduction is provided here to explain how MATE and EEA are used to demonstrate the work in this thesis.

In previous work [124], which forms part of the contribution of this thesis, the author demonstrated how multi-attribute tradespaces could be set up for aircraft and how these could be employed in evolvability analysis (see Figure 2.15). Note that the term ‘scenario’ or ‘timeframe’ is used from now on instead of ‘epoch’, but they refer to the same thing. For this research, as in Lim [7], only two timeframes are considered, i.e. the near-future and far-future. These would coincide with the envisioned entry-into service (EIS) for the first- and second-generation designs. However, note that there could be multiple scenarios per timeframe, which would reflect different ways in which the future may play out. More timeframes could be considered, but for simplicity and demonstration purposes, two sufficed. Other reasons for limiting the number of timeframes are that it would become increasingly difficult to plan much further ahead than two generations and that the kind of substantial design changes for which the techniques in this research were devised are unlikely to happen frequently.

Figure 2.15 shows two MATs – the one on the left represents the near-future EIS, whereas the one on the right represents the far-future EIS. Together, these two timeframes (along with all the scenarios considered for each) and the associated design and tradespaces are collectively called the ‘evolvability space’. Design A in Figure 2.15 is a new design considered for EIS in the near-future scenario. The cost (to the manufacturer) of one aircraft of this design comprises the Research, Development, Test, and Evaluation (RDT&E) cost per aircraft, plus the first-unit manufacturing cost. The manufacturer (OEM) has three options for the

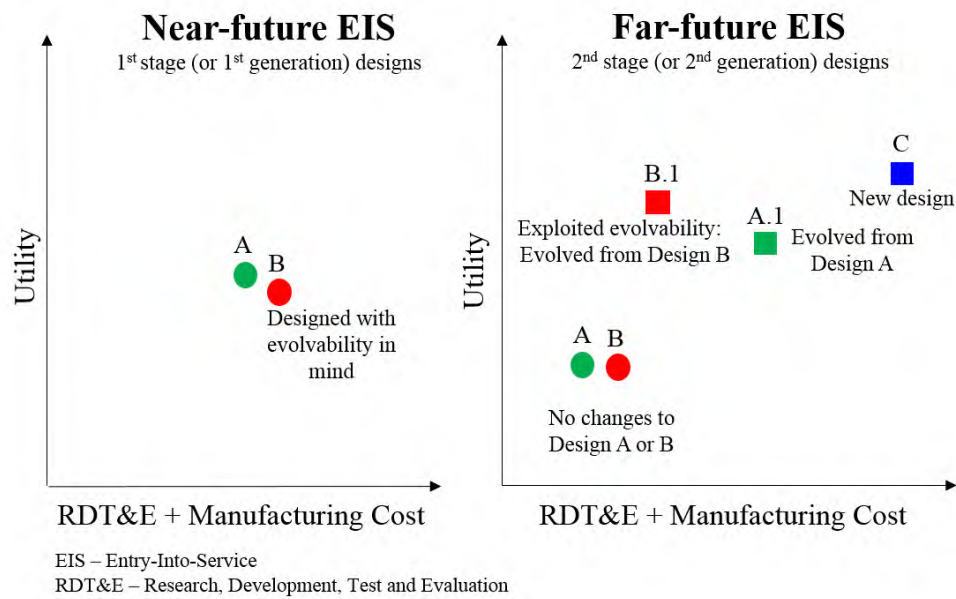


Figure 2.15: Evolvability from the near- to far-future (modified from van Heerden et al. [124]).

subsequent generation designs: completely re-use the first-generation design (i.e. the green dot in the far-future EIS), design a completely new aircraft (Design C), or upgrade the first-generation design (Design A.1). The first option would only be acceptable if the utility of the first-generation design is still above zero in the second-generation timeframe. The cost in this case will only comprise the unit manufacturing cost, which will be less than the first-unit manufacturing cost (when adjusting for inflation). Cost will therefore be low, but the utility probably also. If, on the other hand, the OEM opts for the second option, the full RDT&E cost for a new aircraft, plus first-unit manufacturing cost, will be incurred. The cost will be high, but the requirements could possibly be met exceedingly well.

A more likely course of action would be the third option, which is a compromise. In this case, changes are made to the design that enable it to better meet the new requirements at reduced cost. This entails a combination of re-using components that do not need to change and designing new ones where necessary. As already explained, this provides savings on both RDT&E and manufacturing costs, because only selected components would need to be re-developed. However, this strategy may deliver designs with less performance than what is actually possible with new technologies that have since become available. This is represented by the lower utility of the Design A.1 in Figure 2.15.

To enable further savings in the development and manufacturing cost of future generations, the OEM could invest in designing the first generation to be more

evolvable (Design B). Such a course of action would likely entail a higher first-generation RDT&E per aircraft and first-unit manufacturing cost and lower utility, as compared with designs with no planned evolvability, but may help to increase second-generation utility at less expense (Design B.1).

2.6.5 Process Management

Process management (Phase 5 in Figure 2.8) involves applying procedures that aim to create favourable conditions in which the generation of flexibility in design takes place [80, p. 7]. It addresses management, collaborative engineering, and social aspects and falls outside of the scope of this research.

2.6.6 Critical Discussion: Design for Evolvability

The research in this thesis focuses specifically on contributing to two areas in the phases shown in Figure 2.8 – enabler identification (in Phase 3) and computationally efficient search (in Phase 4). The limitations and associated opportunities related to the techniques in these two areas are discussed briefly in this section. Computationally efficient search is covered first, as the contributions for this area led to exploiting opportunities in enabler identification.

2.6.6.1 Limitations and Opportunities: Computationally Efficient Search

The evolvability exploration methods introduced in Sections 2.3.1 and 2.6.4.3 are powerful techniques to explore evolvability in complex engineering systems. However, as explained in the motivation for this thesis (Chapter 1.2), there are some limitations associated with these, and opportunities abound to exploit their power even further. Two aspects are considered here: the pre-specification of change rules in current techniques, and the disadvantages of available filtering techniques. These are discussed next.

Pre-Specification of Change Rules

All the evolvability exploration techniques described in Sections 2.3.1 and 2.6.4.3 require that the change rules (change mechanisms) between different designs are already specified (which would be the result of Phase 3 in Figure 2.8). For example, in Fulcoly et al. [121] and Fulcoly [120], design for evolvability (DfE) is specified as a design variable and acts as a mass penalty. Redesign cost and schedule time is then calculated as a function of the DfE parameter [120, p. 68].

In another example, in Step 4 of the EvoLVE method from Lim [7] (see Figure 2.12), the possible design elements that may need to be upgraded in the future

have to be identified. The outcome of this is a second-stage design variable vector, after which “scaling laws” have to be developed that model the “physical growth” of the vehicle, along with cost models to quantify the cost of such changes [7, pp. 108-109]. This takes place in Step 6 of the method. After this, when the second-stage design space is populated (Steps 7-9) both the first- and second-stage design variable vectors are used as input (i.e. each second-stage design is linked to at most only one single baseline design). Specifically, for the demonstration of Step 9 of the EvoLVE method, the first- and second-stage design variable vectors are concatenated into a single vector, which is used as input to populate the evolvability space by means of Monte Carlo simulations.

In the motivation of this thesis (Chapter 1.2), it was shown that this manner of populating the far-future design space could prove prohibitive (in terms of computational cost) if it is desired to investigate large numbers of possible near- and far-future designs. Specifically, it was shown that a substantial reduction in computational cost could be realised if the near- and far-future design spaces were to be populated separately first. This would then require some means of assessing similarity across the designs as an estimate for the redesign effort. This similarity assessment will have to be automatic and faster than evaluating a single design point, as, otherwise, the computational cost benefits would be diminished. Further savings is possible, because, if populated separately, the near- and far-future spaces could first be pruned of non-promising designs *before* performing similarity assessment.

Apart from the reduction in computational cost, another benefit would be that the similarity techniques could also be used as an evolvability enabler identification strategy (for Phase 3 in Figure 2.8). In other words, if a technique exists, which could determine automatically which components/parts could be common or non-common, and this is combined with the exploration methods, the designer could focus on ways to increase the commonality, or render non-common items ‘easier’ to change across designs.

Some authors have already proposed to use design/trade space exploration for such a purpose. For example, Khire et al. [125] demonstrated how interactive visualisation could be used to enable a designer to select continuous and discrete variables to be common across a multi-variant product family. This is done by visualising the objective space together with a two-dimensional plot that shows the possible levels of a shared variable for each variant. In Slingerland, Bobuk, and Simpson [126], an approach is proposed in which optimisation is employed together with multi-dimensional data visualisation techniques (specifically parallel

coordinate plots) to enable a designer to select subsets of commonality interactively. Van Heerden et al. [124] proposed a similar technique, where MATE and EEA are used to first narrow down the number of designs to those that perform well in several different near- and far-future scenarios. After this filtering process, parallel coordinate plots are used to select common variable levels across the promising designs. Finally, Roark [127] employed MATE and EEA, along with design structure matrices to facilitate decisions on where in a given system modularity should be incorporated.

The interactive visualisation methods described in the previous paragraph (the first three listed) are useful and powerful. However, only a limited number of variables could conceivably be managed by the designer. Also, in all of the above methods, either only a single system architecture was used, or commonality was only searched for in component pairs that were known to be described by the same set of parameters. The methods will not work as-is for systems with dissimilar architectures/configurations. In addition, in airframes, commonality could exist across segments with complex geometries and mass distributions, such as in the wings and fuselage. None of these techniques can detect such commonality across complex components. Finally, none of them are automatic, which could be problematic when the number of designs, configurations, and scenarios are high. To overcome these issues, there is a need for techniques that could *automatically*:

1. determine which components across two different system architectures are of the same type and connected to other components in the same way, such that they could be send for more detailed similarity assessment; and
2. identify similar segments (based on user criteria) across pairs of components that have complex geometries, complex mass distributions, several attachments to other components, and so forth.

The literature was therefore consulted to search for appropriate techniques from other fields that could be employed or adapted for these purposes. These are covered in Section 2.7.

Disadvantages of Available Filtering Techniques

The changeability metrics discussed above, i.e. filtered-outdegree (FO), value-weighted filtered outdegree (VWFO), Effective (Fuzzy) Normalized Pareto Trace (eNPT/efNPT), Fuzzy Pareto Shift (FPS), and Available Rank Improvement (ARI), could all be employed to filter the evolvability space to find promising designs. However, FO and VWFO only measures the ‘counting’ value of a design (i.e. into how many other designs this design could change) [14, p. 2], and have been superseded by eNPT, efNPT, FPS, and ARI. Of these, FPS and ARI are the most

relevant for this research, as eNPT and efNPT are used for determining the passive robustness of designs (i.e. when the designs do not change).

FPS and ARI only account for differences in the FPN or utility between originating (baseline) and destination designs across epoch shifts and not the utility and cost of the originating designs. Therefore, they cannot be used for filtering purposes on their own. Furthermore, they require a strategy or a specific change mechanism to be selected, and are concerned only with the ‘best’ improvement possible. In other words, for a selected originating design (and strategy or change mechanism), they involve determining the difference between only the maximum utility, minimum cost, or minimum FPN of all the other designs (accessible through a change mechanism) in an epoch and that of the selected design. The ability of the originating design to evolve to other designs (or the ease at which a selected destination design could be created from all of the originating designs) is therefore not accounted for. Therefore, using these metrics as a means of filtering could lead to prematurely reducing the design space too much. Finally, as these metrics focus on changes across specific epoch shifts, they have to be combined in some manner for all the designs and all the epochs to filter the entire evolvability space at once.

Therefore, in light of the above, systematic down-selection techniques are required which could:

1. enable the entire evolvability space to be filtered at once;
2. account for the ability of the originating designs to evolve to all destination designs;
3. account for the ease at which a selected destination design could be created from all of the originating designs; and
4. account for the utility and cost of the originating designs.

A technique that addresses the second and third points could help keep the design space open for longer by preventing designs from being removed based only on ability to evolve to (or from) ‘best’ designs. This is especially important here, because, as pointed out by several authors (e.g. Refs [17], [15], [128], and [129]), it may be advantageous to keep the design space open for longer to delay decisions until further information becomes available. Set-Based Design (SBD) is an approach in which it is attempted to do this. One account of using MATE with SBD was found in the literature (see Junfei [130]). However, in that work, only a single timeframe (i.e. a single MAT) is used. The literature on SBD was therefore investigated in more depth to find appropriate techniques that could be adapted for the purposes listed above (see Section 2.7).

2.7 Similarity Assessment Techniques

As discussed in Section 2.6.6.1, a need was identified for techniques that could i) determine which components pairs across two airframes are candidates for commonality, and ii) could find common segments in these components (which may have complex geometries and mass distributions).

For the first requirement (i), design structure matrices (DSMs) were identified as an appropriate starting point, because these are widely used when the structural layout of a systems architecture is concerned. As will be shown in Chapter 4, a technique was developed that employs design structure matrices and an aircraft object model to partition two given airframes into their constituent components and identify which pairs of components could be dispatched for detailed similarity assessment.

For the second requirement, techniques had to be found that could:

1. find common segments in longitudinally defined bodies, such as the fuselage and nacelle. These bodies may have different lengths and may have components that are attached to them at different locations;
2. determine if two wings could be made common (by using some of the criteria in Section 2.5.5) and, if so, determine what the optimal combination of the wing-tip extension length, chord-extension length, and/or root insert length should be, along with deciding whether the control surfaces are similar enough to be common.

Note that the geometry of other components, such as empennage surfaces, undercarriage, and other supporting structures are usually not defined in detail at conceptual design. It was therefore only attempted to make simple, high-level commonality predictions for these components (see Chapter 4).

The nature of the problem for point 2 above suggests that optimisation might be a suitable approach, and this course of action was pursued (see Chapter 4). Note that, for point 2, intermediate span-extensions (i.e. ‘plugs’ located between the wing root and tip) were excluded from consideration, as these seem to be rarely used in practice (see Chapter 3).

For the first point above, the literature on computational geometry similarity assessment techniques was consulted. This is a vast field and has many applications, including in product design and manufacturing. Many of the techniques are specifically appropriate for feature-based geometry representations. They involve first converting the geometries into simpler abstract shapes, called ‘shape signatures’

[131] or shape representations [132]. After this, the signatures are used for similarity assessment, which usually results in a distance metric for the two.

However, as the bodies (fuselages and nacelles) in conceptual design are usually described with a parametric geometry representation defined along a single axis, and the need here was to also consider other parameters distributed across this axis (such as material thickness and cost) a different approach was sought. Such a technique should specifically be able to find similar segments across the two bodies which might not be aligned. This problem is akin to comparing two documents to find similar text, which is typically required to perform version control and plagiarism detection. Paragraphs may have been inserted, but document comparison software can successfully find which sections of the text are the same across two documents.

One approach often employed in text comparison techniques involves solving what is called the ‘longest common subsequence’ (LCS) problem [133]. This problem is explained here, using the description from Bollobás, Das, and Gunopulos [134]. Consider two sequences $X = x_1, \dots, x_n$ and $Y = y_1, \dots, y_n$. These could represent two strings or numerical data. Note that, if the sequences have different lengths, the shorter one could be padded with ‘dummy’ data to obtain the same lengths. The LCS problems involve finding the longest subsequences in X and Y , $X' = x_{i_1}, \dots, x_{i_l}$ and $Y = y_{j_1}, \dots, y_{j_l}$, respectively, such that:

1. for $1 \leq k \leq l - 1$, $i_k < i_{k+1}$ and $j_k < j_{k+1}$; and
2. for $1 \leq k \leq l - 1$, $x_{i_k} = y_{j_k}$.

To enable higher efficiency, and to allow for searching within a given tolerance (i.e. the values in the sequences do not have to be equal), Bollobás, Das, and Gunopulos [134] propose the following form for the LCS: Let $\delta > 0$ be an integer constant, $0 < \varepsilon < 1$ a real constant, and f a linear function (such that $f(y) = ax + b$), which belongs to the infinite set of linear functions, $\mathcal{L}$. Then, let X' and Y' (in the same form as previously) be the longest common subsequences of X and Y , respectively, such that:

1. for $1 \leq k \leq l - 1$, $i_k < i_{k+1}$ and $j_k < j_{k+1}$;
2. for $1 \leq k \leq l$, $|i_k - j_k| \leq \delta$; and
3. for $1 \leq k \leq l$, $\frac{y_{j_k}}{1+\varepsilon} \leq f(x_{i_k}) \leq y_{j_k}(1+\varepsilon)$.

The LCS problem occurs in many fields. Along with string comparison, discussed above, these fields include bioinformatics (for DNA or protein sequencing) [135], [133], and fields where time-series similarity problems arise, such as in financial markets, telecommunication data, medical signals, and so forth [134].

Many algorithms have been proposed to solve the LCS problem and, for an overview, the reader is directed to Bergroth, Hakonen, and Raita [133]. These algorithms generally fall under the field of dynamic programming in computer science. In dynamic programming, problems are solved by first solving sub-problems and then combining the solutions [135, p. 323]. A combination of two simple algorithms from Cormen et al. [135, pp. 353-355] demonstrate the procedure. The first algorithm creates a ‘recursive array’, which the second algorithm uses to return the actual LCS.

2.8 Set-Based Design

Set-Based Design (SBD), also called Set-Based Concurrent Engineering, is an engineering design and management approach where it is endeavoured to “delay commitment to a particular design alternative in favor of gathering information about the problem” [129, p. 219]. In SBD, rather than following the typical strategy of selecting the ‘best’ design alternative, the approach is to eliminate inferior designs [129, p. 219], [136]. This allows designers to maintain design freedom further into the design process, while, at the same time, increasing knowledge [129, p. 219]. The main benefits of SBD include that it could substantially reduce rework (i.e. work that results as it becomes clear that previous decisions were defective) [137] and that it could greatly improve understanding of the design space [136]. SBD is especially well suited for conceptual design, where it is desired to consider many designs [129].

The origins of SBD as an academic field of study can probably be traced back to the work of Ward, who became aware of the benefits of such an approach and identified and described SBD principles exploited by the Toyota Motor Corporation (see Ward et al. [138], and also [137] for an overview). Sobek et al. [17] expanded on the SBD approach and formulated the following principles:

1. *Map the design space.* This involves defining the feasible regions, exploring trade-offs by considering many alternatives (i.e. designs of different scales and configurations), and communicating options in terms of ‘set of possibilities’ – not single ideas at a time [pp. 73-76].
2. *Integrate by intersection.* As different functional groups gain knowledge on the aspects considered, design teams integrate the designs to create solutions that would be workable for all [p. 76]. This is done through finding intersections of feasible sets; imposing the minimum constraint needed at the time (which ensures flexibility for further adjustments later); and seeking conceptual robustness (creating designs that would work regardless of decisions from other team members) [pp. 76-79].

3. *Establish feasibility before commitment.* This involves “exploring multiple designs in parallel and gradually converging on a single one” [p. 79]. This is performed by gradually narrowing down sets, while increasing detail; staying with the sets once committed; and controlling the process by managing uncertainty at ‘process gates’ (integration events where uncertainty is reduced) [pp. 78-81].

As can be observed from this list, the principles are concerned both with management and engineering practices. The focus here is on the engineering aspects and, in particular, the third principle above, where the gradual narrowing down of sets (i.e. ‘set-reduction’) is of interest. Several techniques have been developed for doing this, including the use of predicate logic to remove infeasible designs [139], interactive visual constraint satisfaction analysis and non-dominated filtering [15], [16], and parametric, energy-based constraint sensitivity analysis along with convex hull techniques [128]. Formal set-reduction techniques have been applied to the design of several different types of systems, including naval ships [136], an airport-city transportation infrastructure system [130], a gearbox transmission system [129], [140], and passenger aircraft [128], [16], [15]. Riaz [15] and Riaz et al. [16] specifically demonstrated how SBD could be used to design aircraft families.

Most of the set-reduction techniques focus on eliminating designs based on their infeasibility. Malak et al. [129] and Rekuc et al. [140] have argued that there is a need for techniques that also take into account the preferences of multiple stakeholders, particularly in the form of multi-attribute utility. They subsequently devised several filtering criteria for SBD, based on multi-attribute utility theory.

2.8.1 Interval Dominance

The simplest criterion proposed by Malak et al. [129] and Rekuc et al. [140] is ‘interval dominance’. Interval dominance could be employed when imprecision in parameters is modelled by means of intervals, which would lead to intervals in expected utility [129, p. 221]. Two possibilities then exist when comparing two design alternatives (see Figure 2.16) – either there would be overlap in the expected utility intervals (Figure 2.16(a)), or not (Figure 2.16(b)). In the latter case, interval dominance applies, i.e. the design with the higher utility interval is preferred and the dominated design could be eliminated. Although interval dominance is simple, it does not capture the full relationship between design alternatives [129, p. 220]. Indeed, a design alternative could still dominate another, even if their respective utility intervals overlap [129, p. 220].

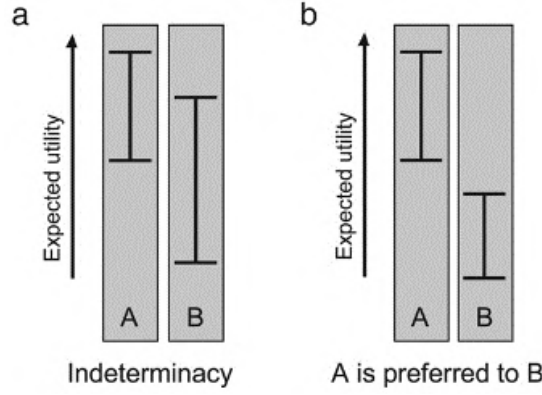


Figure 2.16: Decision possibilities between two alternatives. Case (a) is indeterminate, as the expected utilities of the design alternatives overlap. In Case (b), there is a clear preference for alternative A (from Ref. [129]).

2.8.2 Maximality

In Figure 2.17, it is shown how designs could dominate each other even though their utility intervals overlap. The figure shows that the utility of alternative A is higher than that of B over the entire range of the uncertain parameter. The same applies for design C, which dominates D over the entire range. Design alternatives B and D can therefore be eliminated. Note that alternative E cannot be eliminated as such and a possible criterion for removing it is described in the next section. Malak et al. [129] and Rekuc et al. [140] propose the ‘maximality’ criterion, which was originally introduced by Walley [141], to perform the filtering to eliminate alternatives in a cases such as with B and D. Formally, maximality can be expressed as [129, p. 220]:

$$\min_{\mathbf{y} \in \mathbf{Y}} E[U(A, \mathbf{y}) - U(B, \mathbf{y})] > 0 \implies A \succ B \quad (2.4)$$

In this equation, $\mathbf{y}$ is (in general) a k -dimensional vector that contains the values of k uncertainties; $\mathbf{Y}$ is a set that contains all k -dimensional vectors that are within the domain of imprecision [129, p. 220]. The relationship is therefore that, if the minimum difference between the utility of A and B over the entire imprecise region is positive, then A would be preferred to B [129, p. 220].

Maximality is ‘stricter’ than interval dominance, which means that it could eliminate at least as many alternatives as interval dominance [129, p. 220].

2.8.3 E-Admissibility

In E-admissibility, which was originally discussed by Levi [142], a design is eliminated if, for every point across the uncertain parameters, there is at least one

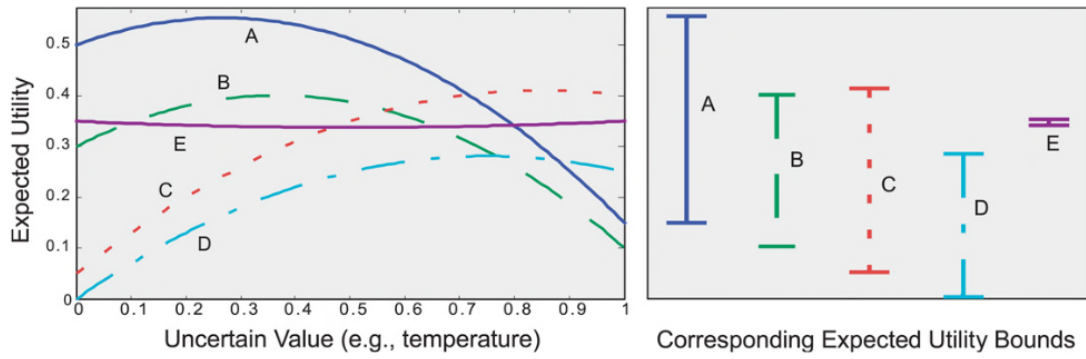


Figure 2.17: The variation of expected utility across the range of an uncertain parameter (from Ref. [129]).

other alternative design that dominates this design in terms of expected utility. This can be explained by considering design E in Figure 2.17. As can be seen, this design is never the worst in terms of utility, but there is always a design that has higher utility (A, B, or C). Using E-Admissibility, design E could be eliminated [140]. Caution must be applied in this case, however. This is because, as can be seen from Figure 2.17, design E could be considered a robust solution in that it performs well across the entire uncertain region, albeit not being the ‘best’. According to Rekuc et al. [140], this is not a problem, if the designer/s is able to resolve the uncertainty before making a final decision. In other words, design E in Figure 2.17 could be eliminated if the uncertainty is resolved before having to commit to either A or C.

2.8.4 Discussion

Because of their incorporation of multi-attribute utility theory, the techniques proposed by Malak et al. [129] and Rekuc et al. [140], discussed above, were deemed particularly appropriate to address the need for systematic filtering techniques identified in Section 2.6.6.1. However, they cannot be used as-is, as they do not account for the multiple timeframes, redesign effort and first-stage cost associated with evolvability studies. The need to adapt these techniques to incorporate these aspects was therefore formulated into the second research question and second objective of this research (see Sections 1.2.2 and 1.3).

2.9 Conclusions

The purpose of this literature review was to introduce the reader to i) the relevant concepts employed in this research; and ii) to highlight the limitations and associated opportunities of current methods in evolvability exploration. The conclusions

of the review are as follows:

1. There appears to be a shortage in the academic literature of collated practical information related to the evolution of passenger transport aircraft. Such information is necessary to achieve the first objective of this research. This is because, to develop similarity techniques that are appropriate and realistic for airframes, an understanding of the actual practices of upgrading/reusing design elements across different aircraft is required.
2. To gain the expected efficiency advantages of evaluating the near- and far-future tradespaces separately, there is a need for techniques that could automatically assess similarity across airframes with dissimilar configurations, as well as finding common segments within the complex constituent components. Using design structure matrices for configuration comparison, optimisation for wing commonality prediction, as well as expressing similarity assessment in body components as a longest common subsequence problem were identified as potential enablers to devise such techniques. An added benefit of these methods would be that they could help designers decide where in an airframe to embed evolvability enablers. This need was reformulated into Research Question 1 of this research: How can similarity between airframes be assessed automatically to predict redesign effort?
3. Although techniques exist for down-selecting designs in MATE and EEA, these tend to focus on finding ‘best’ designs. However, it is often desired to keep the design space open for longer (to maintain design freedom) by removing ‘least desirable’ designs. Set-Based Design filtering techniques were identified as appropriate strategies to enable this to be done in a systematic manner. However, current SBD techniques are not amenable to the multiple timeframes associated with evolvability exploration. Subsequently, there is need to adapt existing SBD methods for this purpose. This need was reformulated as Research Question 2: How can set-based design methods be devised and/or adapted for evolvability exploration studies?

In the next three chapters, it is demonstrated how these limitations and opportunities were addressed. Chapter 3 contains a comprehensive survey of evolvability in civil transport aircraft; in Chapter 4 the proposed similarity assessment techniques are introduced; and Chapter 5 covers the proposed set-based design techniques.

CHAPTER 3

Case Studies in Civil Transport Aircraft Evolvability

Chapter Summary: The purpose of the study presented in this chapter was to obtain a practical understanding of the techniques and strategies used by aircraft manufacturers to evolve their existing designs. To achieve this, single- and twin-aisle jet transport aircraft from three large western manufacturers (Boeing, Airbus, and McDonnell Douglas) were investigated in depth. The academic and industrial literature was combed to find descriptions of design reuse and change across each major model of all three manufacturers. The causes and effects of the changes are explored, and the amenability of the different airframes to change are discussed. The evolution of the payload and range capabilities of the different aircraft was also investigated for each manufacturer. From these studies, it was found that the initial approach to derivative designs appears somewhat ad hoc and that substantial modifications were devised in quick succession to increase both range and capacity. From the 1970s, two distinguishable patterns started to appear – a ‘leap and branch’ and a ‘Z’ pattern. The leaps correspond to major changes, whereas the branches are simple ‘stretches’ or ‘shrinks’. The Z pattern, also documented by other authors, is a progressive increase in range, followed by a simple stretch, and then another increase in range. Design changes were investigated further by grouping them according to the assumed payload-range goals set for the derivatives. Finally, the maximum changes found for salient geometrical design parameters amongst all the aircraft surveyed were documented. These were used as design constraints for the work presented in Chapter 4.

3.1 Introduction

The literature review presented in the previous section resulted in the identification of several methods to design for and explore the evolvability of engineering systems. However, there seems to be a paucity in the academic literature of collated practical information related to the evolution of passenger transport aircraft. Specifically, to achieve the first objective of this research (devise computational techniques that could assess similarity between airframes automatically), it was required to obtain an understanding of the design changes used in practice to evolve the designs of airliners. This was needed to formulate rules for the commonality prediction techniques, which are presented in the next chapter.

Therefore, an investigation was conducted into evolvability practices in actual commercial transport aeroplanes, which forms the subject of this chapter. The purpose was to obtain i) qualitative descriptions of how components, design features, and airframe segments were reused or changed across generations of airliner designs; ii) to determine whether there are any distinguishable patterns in payload-range capability evolution, along with associated airframe changes; and iii) to determine the maximum changes in airframe geometrical design parameters achieved across baseline and derivative pairs. The latter were used in the next chapter as constraints for the techniques developed there.

This investigation formed part of the Descriptive Study I stage of the research, i.e. the results ‘describe’ the phenomenon of evolvability in transport aircraft (see Section 1.5.2). Several sources were employed to conduct this study, including the manufacturers’ websites, books devoted to the history of the aircraft, industry periodicals, three-view drawings, and airport planning manuals. In line with the scope of the research, the study was limited to a selection of civil jet transport aircraft. Only western airliners from the main current and historical OEMs, namely Boeing, McDonnell Douglas, and Airbus, were investigated because there was a reasonable amount of data available for these. Also, only aspects that would normally be considered at conceptual design, i.e. the airframe major components, and the airframe-propulsion integration, were investigated. Freighters and special variants were excluded but might receive mention in the text.

This chapter is organised as follows: Section 3.2 covers Boeing aircraft, Section 3.3, McDonnell-Douglas aircraft, and Section 3.4 is devoted to Airbus aircraft. Section 3.5 covers the history of payload-range capabilities of the manufacturers’ products. In Section 3.6, design changes are categorised by the intended design goal for the descendant aircraft and maximum changes in geometry parameter values are listed in Section 3.7. Conclusions are drawn in Section 3.8.



Figure 3.1: The Boeing model 367-80 (Source: Boeing [146]). The ‘Boeing 707’ marking on the tail is for marketing purposes.

3.2 Boeing

The world’s largest aerospace company [143], Boeing, is one member of a duopoly of large commercial jetliner manufacturers (the other being Airbus). Boeing has a long and illustrious history of producing successful airliners and, as will be discussed below, some of their designs have evolved over decades.

3.2.1 The progenitor – the Boeing 367-80

On July 15, 1954, a brand-new aeroplane with distinctive sweptback wings and four mighty Pratt & Whitney JT3 turbojet engines took to the skies on its maiden flight [144, p. 164]. This aircraft is, of course, the Boeing 367-80 (Figure 3.1). The 367-80 (or ‘Dash 80’, as it was affectionately known) turned out to be of enormous importance in aviation history and had an airframe layout and design features that can still be seen today on thousands of modern passenger and military transport/tanker aircraft around the world. It can be considered as the *de facto* progenitor of all Boeing [145] and (arguably) most other jet transports.

The Dash 80 was built by Boeing using their own funding over the period of about two years to test and demonstrate the technologies required for jet transport and tanker aircraft. According to the legendary Boeing aircraft designer, John E. Steiner [147], the aeroplane “went on to father the KC-135 military tanker and then the long line of 707 commercial transports”. As will be shown, these aircraft went on to serve, to some extent, as baselines for most of the Boeing transports that followed.

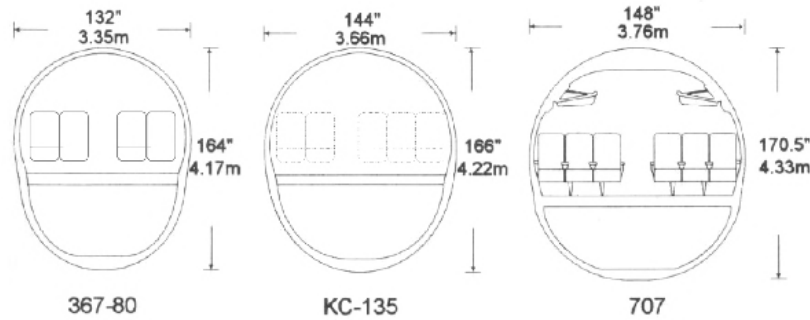


Figure 3.2: The Boeing 367-80 fuselage cross-section evolution [149].

3.2.2 The Boeing 707/720

The changes implemented to the Dash 80 to devise the KC-135 and, subsequently, the 707 were significant. According to Steiner [147], the “KC-135 was a substantially different airplane from the Dash 80” and “gave the engineers a second chance, and as we all know, that can well mean a complete revision”. For example, using the Dash 80 as a baseline, Boeing increased the fuselage width for the KC-135 by about 31 cm (for a total width of 366 cm). This was in response to a requirement from the United States Air Force [145, 148].

The plan was to use this fuselage cross-section on the soon-to-follow commercial 707. Such was the case, until the competition, the Douglas Company, won a major order from United Airlines for their DC-8 in 1955 [147, 148]. The DC-8 was to be introduced with a fuselage width of 373.4 cm, which proved to be the main factor in United’s decision [147]. In response to this, Boeing was forced to increase the fuselage width of the 707 to 376 cm. This meant that the tooling and jigs for the KC-135 body could not be reused on the 707, which increased development cost. However, the wider fuselage allowed for a comfortable six-abreast seating configuration and proved to be popular amongst users. These changes are illustrated by the fuselage cross-section drawings of the 367-80, KC-135, and 707 models in Figure 3.2.

As can also be seen in Figure 3.2, the fuselage cross-sections of the 367-80, KC-135 and 707 are of ‘double-bubble’ (also called double-lobe) design, with a distinct upper and lower lobe. As will be illustrated in the subsections on the Boeing 727, 737, and 757 that follow, the upper lobe dimensions of the 707 were reused on all subsequent Boeing narrow-body designs.

Though the fuselage width was altered substantially, the nose shape and cockpit structure remained little changed across the Dash 80, KC-135, and 707. The nose cone (or radome) was, however, “longer and more pointed” than the Dash 80 [148].

Table 3.1: Major variants of the Boeing 707 family (see p.73 for the 720).

Variant	First flight [150]	Description
707-120	20 Dec 1957	First production 707.
707-120B	22 June 1960	707-120 incorporating turbofan engines and design improvements from the Boeing 720.
707-130		Extended range version of the -120. Fuselage shortened by six frames.
707-130B		707-130 incorporating turbofan engines and design improvements from the Boeing 720.
707-220		Airframe identical to the 707-120, but features more powerful engines.
707-320	11 Jan 1959	Stretched derivative of the 707-120 with longer range. Significantly revised wing along with other airframe changes.
707-320B	31 Jan 1962	707-320 design incorporating turbofan engines. Wing was a significantly revised -320 wing: Krüger flaps added along most of the span; trailing-edge flaps modified near the wing root; different wingtip introduced, which increased span [150].
707-420		Identical to the -320, but with Rolls-Royce engines.

Such reuse of the forward fuselage segment design (especially the nose and cockpit dimensions) across aircraft family members and generations is a widely-used practice. The obvious reason for this ubiquity is that the forward fuselage segment design, manufacturing, testing, and certification processes are usually much more expensive and time-consuming than most of the other fuselage segments. This is because of the more complex geometry of the nose (requiring extensive aerodynamic and structural analysis and testing, as well as expensive tooling); the presence of large ‘cut-outs’ for cockpit windows and doors, incurring structural penalties; and the need for special tests, such as for bird-strikes; amongst others. The aerodynamic advantages that could be gained from revising the nose shape are outweighed usually by the penalties in development cost and time. Indeed, the nose/cockpit design of the 707 was reused to a large extent on the subsequent 727 and 737 families and is still employed on the latest 737s produced today.

Many design updates to the 707 appeared in quick succession in the late 1950s. As will be explained below, it seems that many of these changes were not planned from the outset and rather demonstrated Boeing’s (remarkable) responsiveness to the actions of the competition and the needs of its customers. The major different 707 versions are summarised in Table 3.1 and are discussed next.

The first version of the 707 was the 181 passenger 707-120. Apart from having a

wider fuselage than the 367-80 and KC-135, as discussed above, it was also longer than both. The wing geometries of the KC-135 and 707-120 were almost identical. These were also mostly the same as that of the Dash 80, apart from a five-inch (12.7 cm) wingtip extension on each wing to increase the span to 40 m. However, the wing structure was changed from the KC-135 to better suit airline operations [147] – further reducing commonality with the military model. The -120 and KC-135 wings also featured the addition of retractable leading edge flaps, called Krüger flaps, in between the engines.

Boeing also offered a longer range, ‘short-body’ version of the -120, called the 707-130, of which only Qantas was a customer. The only substantial physical difference between these models was that the -130 fuselage was 3.05 m shorter than the -120 [148]. The lengthening/shortening (referred to as ‘stretching’ and ‘shrinking’) of fuselages by adding/removing fuselage frames (i.e. ‘plugs’) in such a manner has become a very common practice in transport aircraft manufacturing and is employed on most modern airliner families to offer different payload capacities. The ‘tubular’ construction of modern transport aircraft fuselages is of course particularly amenable to this, which is likely one of the main reasons why manufacturers are reluctant to diverge from the tubular concept.

A significant set of modifications to the early 707-120 design resulted in the Boeing 707-320 and -420. These aircraft were the ‘intercontinental’ versions (whereas the -120 was the ‘transcontinental’ version). The only major difference between the -320 and -420 was the engines employed.

The conception of the 707-320 resulted mainly from the need to (again) compete with the DC-8 [147, 148]. The DC-8 design had a larger wing area and higher gross weight than the early 707 designs, which enabled it to cross the Atlantic non-stop — something the 707 could not do [147]. In 1955, Pan Am indicated that they preferred the DC-8, which resulted in Boeing frantically modifying the early 707 design to create an intercontinental version (i.e. the -320).

The -320 was about 2.57 m longer than the -120 [144, p. 169]. It also featured a larger wing area, which was achieved through inserting ‘root insert’ and outboard span-extending plugs [149]. The wingtips were also altered, and the engines placed such that they protrude further forward [148]. The wing planform was further changed by revising the inboard trailing edge through a modification of the rear spar and a different inboard flap arrangement [149]. These changes are highlighted in the comparison of the wings of the Dash 80/707-120 and 707-320 shown in Figure 3.3 (also shown is the wing of the later 720 model, which is discussed later). Other notable airframe changes included a tip-extension for the tail fin to improve yaw stability. This tip extension was made standard and later retro-fitted on all

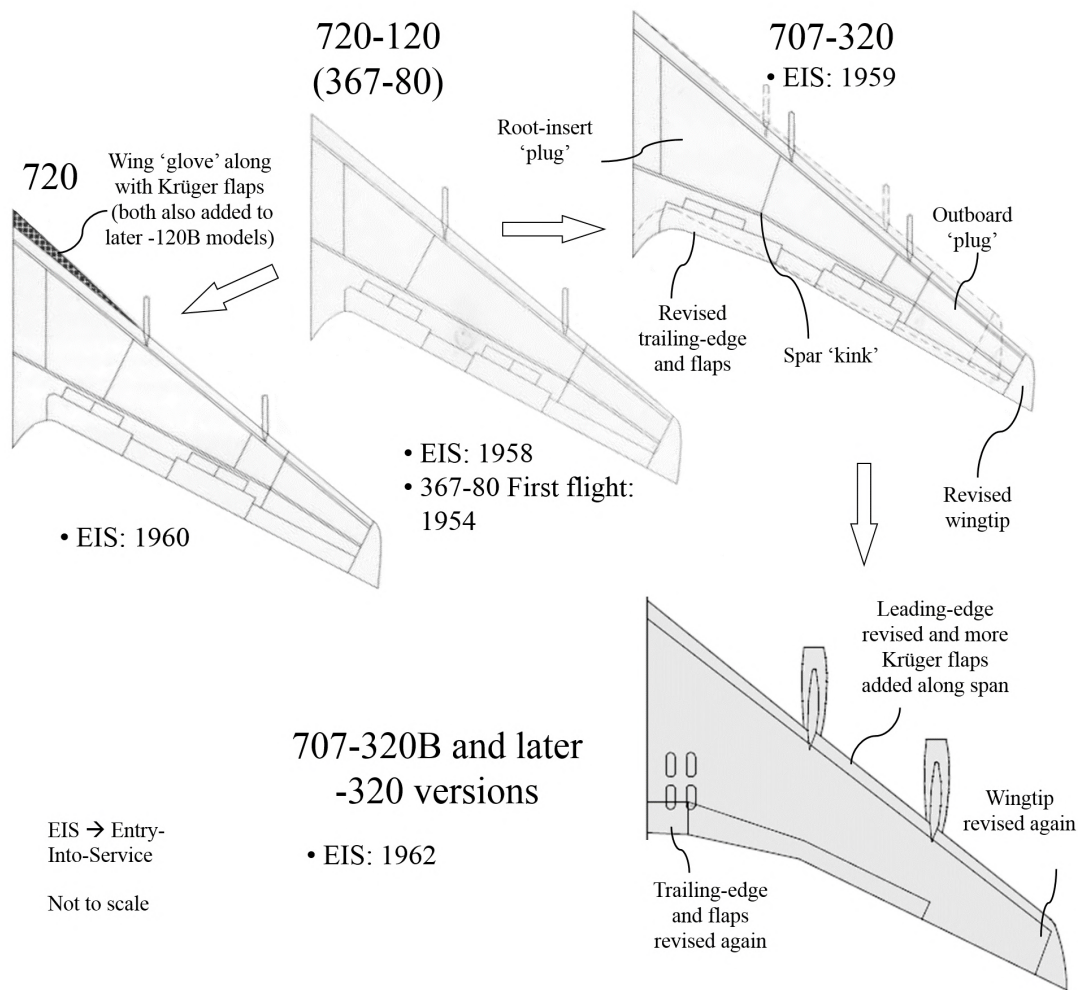


Figure 3.3: Boeing 707 wing evolution (adapted from Almojuela [149] and Boeing CAD drawings [151] (for the -320B), EIS dates from [152]).

707 models [148, p. 38].

3.2.3 The Boeing 720

In 1960, the 707 was followed into airline service by a shorter range, but similar-looking derivative, called the 720 [144, p. 169]. The fuselage of this model was 0.51 m longer than the 707-130 [152].

With the 720 came significant airframe enhancements, especially to the wing. Using the -120 wing as baseline, a 'glove' (a leading-edge extension wrapping around the leading edge of the aerofoil) was added between the fuselage and the inboard engine (see Figure 3.3) [148]. This increased the wing thickness and sweep, and allowed Krüger flaps to be added inboard of the inboard engine [150]. The glove had the effect of increasing the wing area which improved low-speed performance,

whereas the higher sweep allowed for higher maximum speeds [153]. The structure was lighter and the model had lower fuel capacity [144, p. 169].

From 1960, the more efficient and quieter turbofan engine was made available on all 707/720 models. Models which were powered by these were identified by a ‘B’ designation suffix. For example, a 707-120 with turbofans was designated ‘707-120B’. The 707-120B/138B models also received the same aerodynamic enhancements and inboard Krüger flaps as the 720 [150].

Many other ‘smaller’ design changes were applied to the 707/720 airframe during its illustrious career, which included the creation of cargo versions (not covered here). The 707/720 family was very successful, with a total number of 725 commercial models (of all versions) delivered between 1957 and 1994 [154]. Much of this success could likely be attributed to the experience that Boeing acquired with the Dash 80.

3.2.4 The Boeing 727

The 707 was followed into service by a more ‘unconventional’ configuration (for the time), called the 727, which first flew on November 27, 1962 [155]. With 1,832 aircraft delivered between 1964 and 1984, the 727 was even more successful than the 707. It was a smaller aircraft and was designed to operate from relatively short runways and on shorter routes. Like the 707 and 720, it had highly swept wings (although the wings were of a new design), but, instead of the conventional empennage (i.e. low-mounted horizontal tail), it donned a T-tail, along with three fuselage-mounted turbojets. The 727 therefore had a distinctively different airframe configuration than the 707 and 720.

Yet, some of the design was still noticeably common with that of the 707/720 – they had nearly identical nose sections. The fuselage cross-section above the floor (i.e. the top lobe) was also kept the same. The 727 fuselage cross-section was also employed on its intended replacement, the Boeing 757, which was introduced into service in 1983. The 757, however, used a new nose section to increase commonality with the Boeing 767, which was developed in parallel with it. This is discussed later (see Section 3.2.7). The variants of the 727 are shown in Table 3.2.

The first model of the 727 was the -100 model, which was powered by three Pratt & Whitney JT8D engines. To offer increased capacity, Boeing introduced a stretched version, the 727-200, in 1965.

From 1972, an improved -200 version became available, which was designated ‘Advanced’. The Advanced 727 had more powerful engines and structural design changes (see Table 3.2). These enabled more fuel to be carried, which increased

Table 3.2: Major variants of the Boeing 727 family.

Variant	First flight	Description
727-100	9 Feb 1963 [156]	Basic original version.
727-200	27 July 1967 [156]	Stretched derivative of the -100.
727-200 Advanced	1971/72 [156]	Improved 727-200. Thicker wing skins and strengthened undercarriage [144, p. 177].

range.

As a family of aircraft, the 727 had seen fewer major airframe geometrical changes than the 707/720 family, with only the length of the fuselage changing substantially. The original wing design (especially the high-lift devices) simply sufficed and allowed substantial increases in MTOW. This, in turn, allowed stretching for higher payload capacity, as well as increased range.

3.2.5 The Boeing 737 Family

The 737 was designed in the early sixties for short to medium ranges and to land on runways less than 1830 m [157, p. 4]. From the beginning, it was aimed to reuse as much as possible of the design of the 727 [158, p. 12]. This would save time and money and make it attractive to existing 707/727 users [158, p. 12]. However, unlike the 707/720 or 727 families, the 737 employed two engines that were wing-mounted. Its configuration was therefore again different from its predecessors.

Along with continuously improving propulsion, the 737 airframe design has undergone significant evolution, with several modifications applied to the fuselage, wings, empennage, and nacelles. The latest version, called the 737 MAX 10, was introduced in 2017 [159] and, although a very different aircraft from the first 737-100, still retains many of the same design features such as the basic fuselage geometry.

The decision to place the engines under the wing was a significant one. Boeing found that using this arrangement would enable the use of a short, wide body and a low horizontal tail [147, p. 14]. The same upper fuselage cross-sections as the previous Boeing jetliners could therefore be employed [152, p. 72]. This was a major advantage, as the aircraft would have the same passenger and cargo configuration of the 707 and DC-8 [147, p. 14]. Subsequently, the 737 design could retain a 60% parts commonality with the 727 and 707, predominantly in the cabin and flight deck [145, p. 111].

From an evolvability point of view, another advantage of the engine-under-wing configuration is that the engines are close to the centre of gravity (CoG). This

Table 3.3: Major variants of the Boeing 737 Original family.

Variant	First flight [160]	Description
737-100	9 April 1967	Original version.
737-200	8 Aug 1967	Similar to 737-100, but fuselage lengthened (plugs: 0.91 m fore, 1.02 m aft) [157, p. 5].
Advanced 737-200	1971 [160]	Dimensionally the same as -200 [157, p. 4], but with more powerful engines and higher fuel capacity [144, p. 183]. Featured the use of composite materials in the interior, rudder, elevator, and ailerons [160]. Subsequent improvements applied to increase range and payload. This included the addition of an auxiliary fuel tank and increased gross-weight [160].

allows heavier engines to be introduced without a large shift in the CoG, which would inevitably occur if heavier engines were to be attached to the rear fuselage [69]. Furthermore, because the wider fuselage enables more seats abreast, a relatively smaller amount of stretching is needed to increase the passenger capacity than would be with fewer seats abreast. Finally, growth was perhaps also made easier by both spars of the 737 wing being kinked. This arrangement provided more room for fuel, which could be exploited later to increase range. Much of the success of the derivatives could probably be attributed to these aspects ([145, p. 110] quoting Jack Steiner).

The 737 family can be divided into four generations. These are the ‘Originals’; the ‘Classics’, the ‘Next-Generation’ (NG); and the ‘MAX’. Each of the generations consisted of multiple members, with different payload and range capabilities and are discussed next.

3.2.5.1 Boeing 737 Originals

The first 737 was the -100 model. It performed its maiden flight on 9 April 1967 [159] and had room for about 85 passengers in a two-class configuration [157, p. 13]. The -100 and all subsequent ‘Original’ variants were powered by versions of the Pratt & Whitney JT8D turbofans [152, p. 27]. These aircraft also employed flaps and leading-edge devices that were similar to those on the 727 [144, p. 183].

Soon after introducing the 737-100, Boeing started development of the 737-200 as a response to the emerging Douglas DC-9-30 [147, p. 14]. Apart from a longer fuselage, all dimensions remained identical to those of the -100 [157, p. 4]. A further improved variant, called the 737-200 Advanced, was later introduced [160]. The 737 Original family members are summarised in Table 3.3.

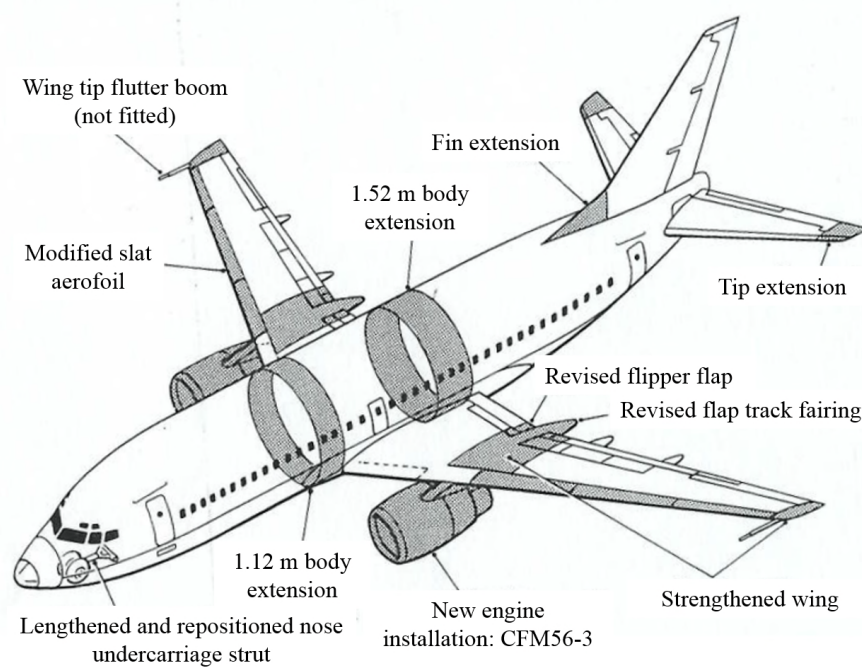


Figure 3.4: Boeing 737-300 changes from the 737-200 (adapted from Wright [158, p. 29]).

3.2.5.2 Boeing 737 Classic

With development starting in 1979, the 737 ‘Classic’ family was pursued by Boeing to exploit the much-improved fuel-consumption and low-noise characteristics of the then new CFM56-3 turbofans [152, p. 74]. The first of the Classic members to be developed was the 737-300 (Figure 3.4), which first flew on 24 February 1984 [161]. It featured a stretch of the -200 fuselage which allowed a maximum seating capacity of 156 passengers [152, p. 74].

The wing was changed substantially. According to Wright [158, p. 26], a wingspan extension using a new centre-section would usually be considered for larger and heavier engines. However, with wing-mounted engines, this would have meant a complete redesign. Boeing wanted to avoid this and resorted to only adding 23 cm span extensions at each wingtip [152, p. 74]. Despite this, from early calculations it was shown that the smaller increase in wing area would not be enough to keep the higher approach speeds (resulting from the higher design weights) within acceptable limits [158, p. 26]. Therefore, a new leading-edge slat was integrated from the engine pylon to the wingtip. The new slat increased the chord over the whole wing by an average of 4%. The resulting larger wing area ensured that the increase in approach speed remained below 5 kt [158, p. 29]. The trailing edge flaps were also revised where the -100/-200 engine protruded aft of the wing. So-called ‘flipper flaps’ were added at this location and the flap track fairings were changed

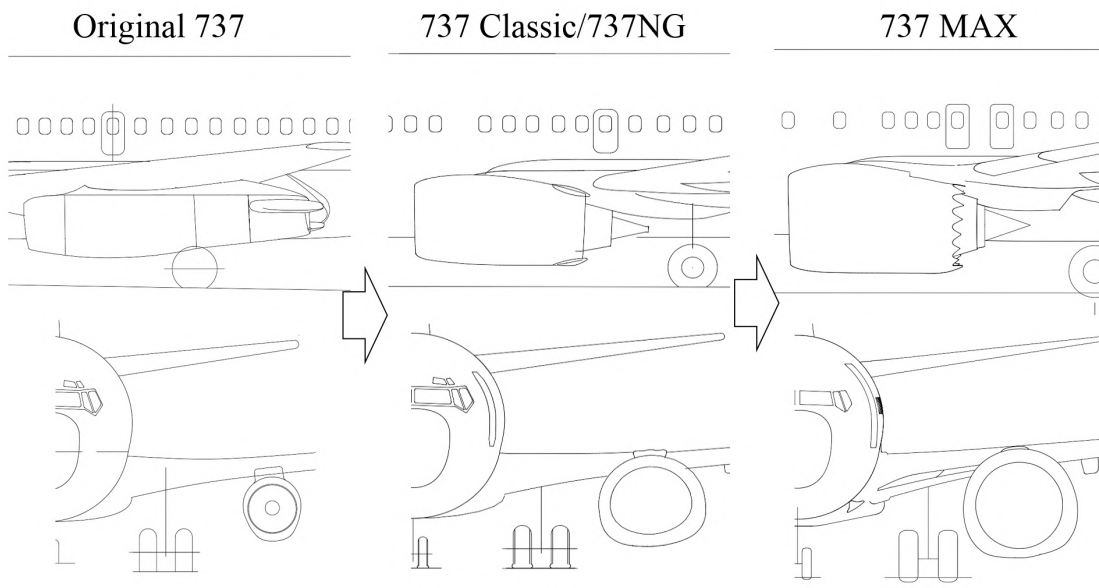


Figure 3.5: Boeing 737 nacelle evolution (created from Boeing CAD drawings [151]).

[152, p. 74]].

Empennage modifications included an increase in the vertical stabiliser area [152, p. 74], through the addition of a triangular wedge at the lower leading-edge. The horizontal stabiliser span was increased using tip extensions [157, p. 5], along with an increase in elevator span [158, p. 29].

The nose wheel leg was lengthened for increased engine-ground clearance [158, p. 29]. The main undercarriage could also have been extended to make room for the larger engines, but this would have implied a redesign [158, p. 28]. Therefore, to ensure proper clearance, the engine nacelle was made flat at the bottom and the accessory drives were placed on the side of the engines [152, p. 74] [158, p. 28]. The engine was also attached to the pylon such that most of the assembly protruded forward of the leading edge [158, p. 28]. Subsequently, the sections attached to the pylons had to be strengthened. To prevent overheating of the bottom wing surface and trailing edge flaps, the engine was tilted at an angle to deflect the exhaust downwards [158, p. 28]. These modifications solved the clearance problem and provided more room for fuel because the area above the engines could not be used to accommodate fuel in the earlier versions [158, pp. 28-29]. The engine increased in size again with the introduction of the latest version, the 737 MAX (see Section 3.2.5.4). The evolution in the engine-wing integration is illustrated in Figure 3.5.

These changes were substantial, but, according to Norris and Wagner [145, p. 113], “it was the inherent strength of the original design and the positioning of the

Table 3.4: Major variants of the Boeing 737 Classic family.

Variant	First flight	Description
737-300	22 Jan 1998 [161]	Stretched derivative of the 737-200 (plugs: 1.12 m fore, 1.52 m aft) [152, p. 74]. See text for other airframe and propulsion changes.
737-400	19 Feb 1988 [162]	Stretched version of the 737-300 (plugs: 1.83 m fore, 1.22 m aft) [157, p. 5]. Additional emergency exits. Krueger flaps added outboard of the engine pylons [152, p. 74]. Uses more powerful version of the CFM56-3 [158, p. 53]. Apart from the fuselage plugs and local strengthening of the structure, the -300 and -400 are almost identical [145, p. 115].
737-500	20 June 1989 [163]	Fuselage same length as 737-200 [158, p. 53]. Structurally essentially the same as the -300 and -400 [145, p. 116].

engines beneath the wing rather than at the tail that allowed the new developments to take place.”

3.2.5.3 Boeing 737 Next-Generation

In 1991, Boeing started working with more than 30 airlines to develop further derivatives of the highly successful 737 [159]. This led to the 737 Next-Generation (737NG) being launched in June 1993 [159]. Boeing capitalised on lessons learned on the then-recent development of the Boeing 777. The resulting product was a new aircraft family in which “20% of airframe spares and 70% of ground support equipment remain common with earlier models” and, within the NG family, 95% of spares are common [164, p. 70].

The changes to the ‘Classic’ family were substantial, however. First, the wing area and fuel capacity were increased by 25% and 30% respectively. This was done by means of a 43 cm chord increase [159], [164, p. 70] and a 272 cm semi-span extension at each tip [164, p. 70]; These changes modified the wing’s aerofoil sections. In addition, the CFM56-3 turbofan, used on the Classic variants, were replaced by versions of the more advanced CFM56-7 [157, p. 7]. Blended winglets were made available as an option on later NGs and as retrofit on in-service aircraft [165].

Other modifications included [164, p. 71]: new double-slotted flaps; a new wing-body fairing; upgraded and simpler undercarriage; horizontal tail tip-extensions, a longer elevator; fuselage strengthening applied to handle the larger tail loads and higher design weights; as well as the addition of a root-insert to increase the vertical stabiliser height to 7.17 m. The variants of the 737NG family are summarised in

Table 3.5: Major variants of the Boeing 737NG family.

Variant	First flight [159]	Description [157, p. 6].
737-600	22 Jan 1998	Same fuselage dimensions as the 737-500.
737-700	9 Feb 1997	Same fuselage dimensions as the 737-300.
737-800	31 July 1997	Stretched 737-400 fuselage.
737-900	3 Aug 2000	A stretched derivative of the 737-800 (1.37 m plug forward, 1.07 m plug aft).
737-900ER	1 Sep 2006	Increased capacity, long-range version of -900. Strengthened undercarriage; heavier wing skin gauge; flat aft pressure bulkhead; and additional emergency exits [159].

Table 3.5.

3.2.5.4 Boeing 737MAX

Boeing launched the ‘737 MAX’ family as a new-engine variant in August 2011 [166]. The engine to be used was to be the CFM Leap-1B [167] and a new nacelle and pylon were designed to accommodate it [168]. The engines are cantilevered higher and further forward than with the NG [169] to retain ground clearance. The nose gear strut was lengthened by 20 cm for the same reason [169]. This necessitated some changes to be made in the lower nose where the nose gear is housed [169].

Apart from the new engines, the MAX family is fitted with new laminar flow wing-lets, which are of a distinctive ‘split-tip’ type [170]. Other improvements include a re-contoured tail cone to reduce drag, along with numerous other small aerodynamic enhancements [168]. The main landing gear was reinforced, and the fuselage skin was made thicker at some locations [168]. Combined, these improvements reduce fuel consumption by 14% over the 737NG [168].

The MAX is offered in four versions (see Table 3.6). These are essentially direct replacements for the NG versions, but the MAX 7 is slightly longer than the -700 and a new capacity variant was introduced as the ‘MAX 10’.

3.2.6 The Boeing 747

The gigantic Boeing 747 was designed to meet rapidly growing traffic. The go-ahead for the 747 was given on April 15, 1966 [147, p. 16] and the first flight was on the 9th of February 1969 [152, p. 76]. The different variants are summarised in Table 3.7, followed by in-depth discussions on each. Figure 3.6 highlights the substantial major changes across the variants.

Table 3.6: Major variants of the Boeing 737 MAX family.

Variant	First flight	Description
MAX 7 (737-7)	16 Mar 2018 [171]	Slightly longer fuselage than the 737-700 [167] (plugs: 1.17 m fore, 0.76 m aft) [172]. Employs MAX 8 wing and landing gear (to handle higher weights) and features a pair of emergency exits over the wing rather than one. Structural ‘re-gauging’ (thickening) and other strengthening applied [173].
MAX 8 (737-8)	29 Jan 2016	Major derivative of the 737-800 [167]. See text for details.
MAX 9 (737-9)	13 Apr 2017	Derivative of the MAX 8 (fuselage based on the 737-900ER). [167]
MAX 10 (737-10)	Launched 19 June 2017 [174]	1.67 m stretched derivative of the MAX 9 [174]. According to Boeing [175], it will have new levered main undercarriage to ensure adequate margin for rotation at take-off, new emergency exits, new rear pressure bulkhead, and a revised wing.

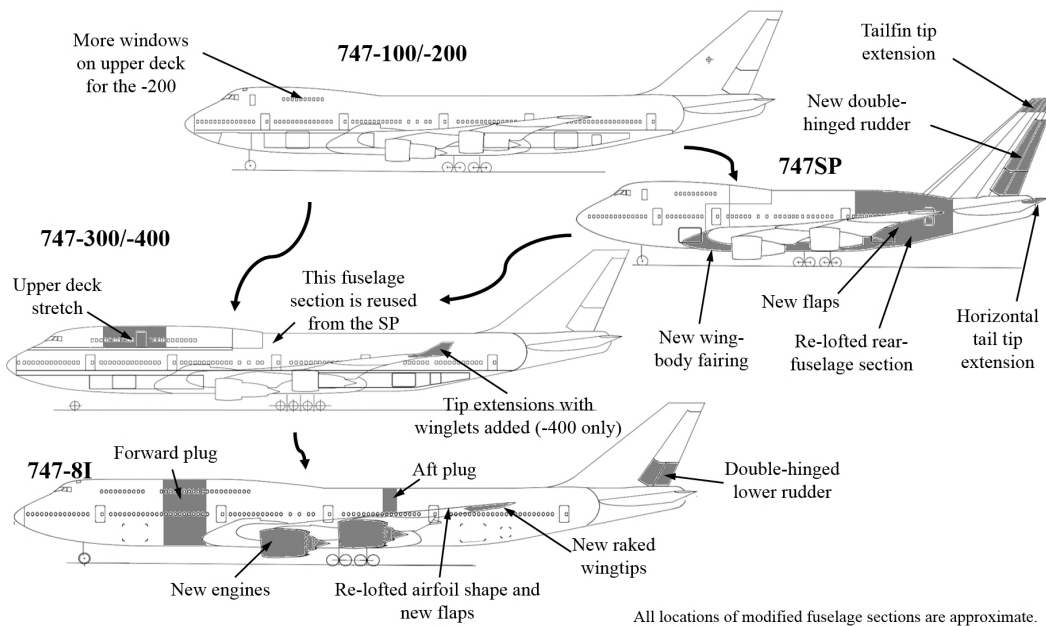


Figure 3.6: Major changes across Boeing 747 variants (created from Boeing CAD drawings [151]).

Table 3.7: Major variants of the Boeing 747 family.

Variant	First flight	Description
747-100	9 Feb 1969 [152, p. 76]	Original basic version.
747-100SR	Introduced: 1973 [152, p. 76]	Short-range (‘SR’) variant of the -100. Strengthened airframe for more frequent landings [144, p. 202].
747-200B	11 Oct 1970 [176]	Higher payload/longer-range derivative of the -100. Overall dimensions same as -100. Stronger structure, with thickened skins. Strengthened wing spars, landing gear beams, flaps, and rib-and-wing panel splices [174]. The fuselage has a strengthened keel beam, gear supports, stringers, skin, and door frames [174]. Local reinforcement of the landing gear was applied and new nose tires were employed [174]. The empennage was made stronger by making changes to the torque box and centre section [174].
747SP	4 July 1975 [177]	Long-range, shorter-body variant of the -100. See text for details.
747-300	5 Oct 1982 [152]	Derivative of the -200 with stretched upper deck, but with similar gross weights and engines [152, p. 78].
747-400	29 Apr 1989 [178]	Derivative of the -300 with significant changes made to structure and systems. The wing-body faring was recontoured and an extra wing leading edge flap was added [174]. A fuel tank was incorporated in the horizontal stabiliser [174]. Wingtip extensions were fitted along with new winglets [145, pp. 140-141].
747-8	March 2011 [179]	Major derivative of the -400. Stretched fuselage and substantially revised wing, along with other modifications (see text for details).

3.2.6.1 747 Classics

The first 747s were the 747-100 and 747-100SR. These were soon followed by the 747-200B. The 747-200B was almost identical to the basic -100, but carried more payload, featured longer range, and had more powerplant options.

The 747SP (Special Performance) was a long-range variant of the 747-100, with the fuselage shortened by 14.6 m [152, p. 76]. It has the same tank capacity as the -200B but accommodates only about 288 passengers [152, p. 76].

The shrink was not entirely straightforward. A fuselage section had to be removed forward of the front spar, as is usual for shortening a fuselage, but the upper section just aft of this had to be removed as well, to make room for the rear part of the hump there [145, p. 140]. A revised centre section, to lower mass, along with a

new, smaller wing body fairing was also incorporated. To replace sections that were removed ahead of the rear-pressure bulkhead, a new aft fuselage section had to be designed [145, p. 140]. In addition, the fuselage was made lighter in certain areas.

The tailfin was about 1 m lower at its base (because of the new aft fuselage section), but received a 1.52 m tip-extension and a double-hinged rudder to account for the smaller moment arm arising from the shorter fuselage [145, p. 140]. The horizontal stabilizer span was also extended by 1.52 m [145, p. 140] through tip extensions.

The wing remained similar to that of the 747-100. However, the structural gauges of the wing-box and carry-through structure ribs, spars, and stringers were reduced to make the wing lighter [145, p. 140]. The trailing-edge flaps were replaced with simpler, single-slotted flaps [145, p. 140].

3.2.6.2 747-300 and 747-400

By the late seventies, stretching of the 747-100/-200 had been considered but, eventually, only an upper deck extension of 7.11 m was adopted by 1980 [152, p. 78]. The new version to incorporate this longer upper deck was called the 747-300, which first flew on the 5th of October 1982 [152, p. 78].

A significantly improved derivative of the 747-300 was launched in October 1985 [152, p. 78]. This aeroplane was designated the 747-400 and much of the experience gained with the then recently developed Boeing 757 and 767 was employed in its development [145, p. 140]. Although having the same overall dimensions (except for span), major changes were made to the -300 Baseline. These involved the addition of winglets along with substantial revision of the structure and aerodynamics (see Table 3.7). The winglets improved aerodynamics, while keeping the span within operational limits [180, p. 72]. Despite the increased span, the use of new alloys in the wing structure made it about 11% lighter than the -300 wing [180, p. 73]. These modifications improved fuel economy and lowered operating costs, and the -400 became the best-selling version of the 747 [145, p. 141].

3.2.6.3 747-8I

A major new variant of the 747, the 747-8, was announced on November 14, 2005 [179]. The ‘-8’ indicates that the new 747 bears a strong relation with the 787 – it employs versions of the 787’s GEnx engines, the 787’s ‘sawtooth’ nacelles, and the 787’s raked wingtips [179]. This is an example of mimicry (see Chapter 2.5.5).

The -400 fuselage was stretched by means of inserting two plugs (4.10 m in front of the wing and 1.5 m aft). This was the first time the 747 was stretched and, at

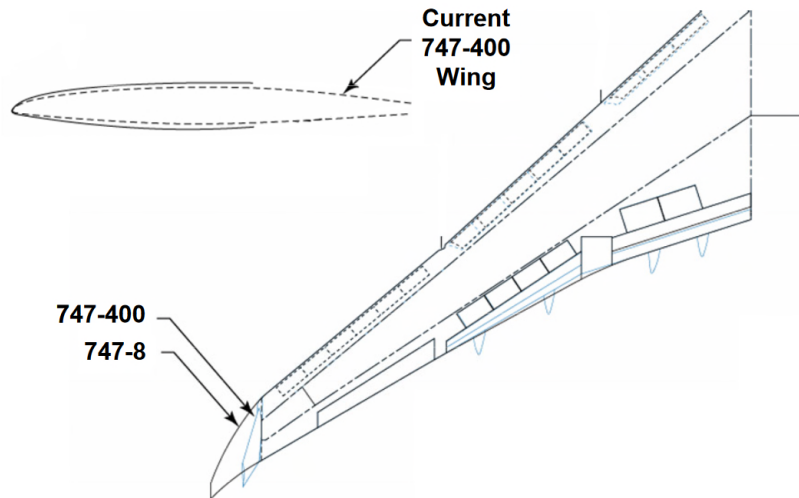


Figure 3.7: Differences between the 747-400 and 747-8 wings (from Collisson [181]).

76.3 m, currently makes it the longest commercial passenger aircraft in the world [179].

With the introduction of the -8 also came the most substantial modifications to the original 747 wing. The outboard wing was relofted, i.e. the aerofoil shape changed, but the basic structural layout was kept the same [181]. This new loft smooths into the original shape where the wing meets the fuselage in order to keep the centre-section geometry the same [181]. The relofting improved L/D and altered the lift-distribution, which helped improve structural efficiency [181]. Thicker aerofoil sections were employed, which did not impose a drag penalty, because of advances achieved in aerodynamics since the original 747 was introduced [181]. The flaps were changed to double-slotted inboard and single-slotted outboard and redesigned flap tracks and fairings were fitted [181]. As already mentioned, the 787 style raked wingtips were adopted as well [181].

Other significant changes included new main undercarriage wheels, tires, brakes, and trucks [181]. A double-hinged lower rudder was introduced to increase yaw control authority. Materials from both the 777 and 787 programmes were adopted [181].

The 747-8 is a very modern aircraft, yet much advantage was gained by deriving it from the 747-400 (and taking advantage of lessons learned from the 777 and 787 programmes).

3.2.7 The Boeing 757 and 767 Family

Design for a clean sheet aircraft, the twin-jet widebody Boeing 767, commenced in mid-1978 [147]. At the same time, the ‘757’ designation was reserved for a slightly smaller aircraft with shorter range [147], which was to replace the 727. The 757 was finally given the go-ahead in 1979 [147].

Commonality between the 757/767 was an important factor in the design. According to [144, p. 246], Boeing claimed that the 767 is “42.8 percent identical to the 757, [and] 19.7 percent similar.”

The wing designs of these aircraft were provided with substantial planform area to ease stretching of their respective fuselages in the future [182]. The 757 wing root-chord was also wider than needed, which allowed accommodating unusually long landing gear. The longer gear increased clearance for larger engines but also left ample margin for stretching the fuselage [145, p. 150]. This decision was based on lessons learned from the 707, where the gear of that aircraft was designed to be short to save weight but resulted in the 707 not being adequately amenable to much stretching [145, p. 150]. For the same reasons, the kink in the wing of the 767 was further outboard, which increased wing area and left adequate margin for growth [145].

3.2.7.1 757

The 757 employed the fuselage cross-section of the 727 [152, p. 80], but had a new wing. Originally, the design also incorporated the 727 nose, but, because of the decision to maintain cockpit commonality with the 767 [152, p. 80], the windscreen was changed to a curved design. This necessitated a new tapered forward fuselage section between the flight deck and constant fuselage sections [182]. In addition, the nose shape was now completely new and unique amongst Boeing jetliners.

Early concepts for the 757 also sported the T-tail of the 727, but the advantages to be gained from commonality were nullified as the new aircraft’s stability and control requirements diverged from the previous design. The horizontal tail was therefore moved to the more conventional lower location, to increase commonality with the 767 [182].

There were two major variants of the 757, the 757-200 and 757-300, which are listed in Table 3.8. There was also an earlier 757-100 design, but this was abandoned.

Table 3.8: Major variants of the Boeing 757 family.

Variant	First flight [183]	Description
757-200	9 Feb 1982	Basic variant.
757-300	2 Aug 1998	Stretched 757-200 (plugs: 4.06 m fore, 3.05 m aft) and strengthened wings, pylons, undercarriage, and new wheels [184].

Table 3.9: Major variants of the Boeing 767 family.

Variant	First flight	Description
767-200	26 Sep 1981 [185]	Basic model.
767-200ER	6 Mar 1984 [185]	Derivative of the -200. Centre-section fuel tank added.
767-300	30 Jan 86 [185]	Stretched derivative of the 767-200ER (plugs: 3.07 m fore, 3.35 m aft) [152, p. 85]. Parts of wing and fuselage skin thickened and undercarriage legs strengthened [152, p. 85].
767-300ER	9 Dec 1986 [185]	Higher gross-weight variant of the -300, with additional fuel capacity and airframe strengthening.
767-400ER	9 Oct 1999 [185]	Stretched derivative of the 767-300ER (plugs: 3.36 m fore, 3.07 m aft) [185]. Strengthened airframe [185] and longer main landing gear. 767-300ER engines, nacelles, and struts employed [186]. New, raked winglets. Strengthened wing box (thicker ribs, spars and skin) [186][185]; new main undercarriage featuring longer legs and new brakes and new tyres [186],[187]. Fuselage structure strengthened to withstand higher bending loads [187].

3.2.7.2 767

The 767 first flew on 26 September 1981. The 767 came in several versions, which are summarised in Table 3.9 and the text below. Originally, (as with the 757) a shorter fuselage length option, the 767-100, was offered, but, owing to a lack of orders, this model was dropped [152, p. 82].

The first 767 was the 767-200. An extended range (ER) version followed, which was created by increasing the gross-weight and modifying the carry-through structure to act as a fuel tank [182]. This enabled the -200ER to fly on transatlantic services. The next variant, the 767-300, was a stretched version of the -200, with a slightly strengthened airframe. With the later 767-300ER, extra fuel capacity was made available in the centre section, and higher thrust engines were incorporated [185]

[152, p. 85]. The gross weight was increased again for this variant and further structural reinforcement was applied [185]. The final, longer-range 767 variant, the 767-400ER, is a stretched derivative of the -300ER. It features new raked winglets which extended the wingspan [186],[187].

3.2.8 The 777 Family

On the 29th of October 1990, Boeing launched a brand-new aircraft, designated the Boeing 777. A new circular fuselage cross-section with a diameter of 6.2 m was adopted [188, p. 35]. This, circular cross-section was different from all preceding Boeing jetliners. However, the nose geometry of the 767 was largely retained [189].

Norris and Wagner observe that the wing was large, with substantial room for fuel [188, p. 36]. The large distinctive six-wheel bogie main undercarriage design allows the aircraft weight to be spread more evenly on runways and taxiways [p. 40]. The lower ‘pavement loading’ would allow for growth versions, without having to add an extra leg, as was needed for the DC-10-30 when it was derived from the DC-10-10 baseline [p. 40]. However, the nose gear design remained similar to that of the 767 [p. 41].

Although being a new aeroplane, the valuable experience gained from previous projects was exploited fully. For example, the Structures Chief Engineer on the 777 programme, Larry Rydell (as quoted by [145, p. 170]), described the 777 wing box as a “carbon copy of the 767 philosophy”. This is another example of mimicry.

The Boeing 777 has become very successful and several variants exist and are in development (see Table 3.10). The 777-200 was the basic medium-range version and the -200ER is the extended range version. The 777-300 was launched on the 26th of June 1995 and features a 19-frame stretch of the 777-200 fuselage [190]. According to a former -300 programme manager, Jeff Peace (as quoted in [145, p. 173]), the 19-frame plug was decided on, as it provided a satisfactory trade-off between take-off rotation angle and take-off speed. This enabled the same main undercarriage to be employed.

Ultra-long-range versions were studied by Boeing from the start of the 777 programme [189] and eventually led to the combined launch of the 777-300ER and 777-200LR on the 29th of February 2000 [190].

Both the -200LR and -300ER have increased fuel capacity with added centre and rear-cargo hold fuel tanks [190]. The -200LR has the same fuselage length as the -200 and -200ER, whereas the -300ER has the same length as that of the -300. External airframe changes were made, however, the most discernible being the addition of ‘raked’ wingtip extensions to both variants [190]. Furthermore, more

Table 3.10: Major variants of the Boeing 777 family.

Variant	First flight	Description
777-200	12 June 1994 [190]	Original basic variant.
777-200ER	7 Oct 1996 [190]	Extended range derivative of the -200. Additional fuel capacity in the centre-section tank. Strengthened wing, fuselage, undercarriage, and engine pylons.
777-200LR	15 Mar 2005 [191]	Ultra-long-range derivative of the -200ER. More fuel capacity, raked wing-tips added, stronger structure, and higher thrust [190].
777-300	16 Oct 1997 [190]	Stretched derivative of the -200/-200ER (plugs: 5.33 m fore, 4.8 m aft) [190]. Strengthened inboard wing and landing gear [190]. Airframe skin and keel beam strengthened [145, p. 173] to endure higher body-bending loads.
777-300ER	24 Feb 2003 [190]	Long-range derivative of the -300. Strengthened airframe, more fuel capacity and raked wingtips added. Substantial revision of the undercarriage (see text). Higher thrust.
777-8	N/A	Shorter derivative of the 777-9.
777-9	N/A	Major stretched derivative of the 777-300ER. New wing and other substantial revisions to the airframe, along with new engines.

powerful versions of the GE90 powerplant are employed on both the -200LR and -300ER [190].

Because of the combination of the longer fuselage and high operating weight, the -300ER features more substantial revision to the main landing gear than the -200LR. It incorporates a semi-levered actuator that maintains the bogie-fuselage angle in such a manner that the aircraft can pivot on the rearmost wheels at rotation during take-off (see Figure 3.8). This increases the allowable rotation angle, which lowers the required take-off speed and distance [192]. In addition, the nose gear on the -300ER had to be made extendable [190].

A substantial 777 derivative programme, the ‘777X’, was formally launched in November 2013 [193]. The new family would consist of the 400-seat ‘777-9’, the long-range ‘777-8’, and the ‘777X Freighter’. The 777-8 will be a replacement for the 777-200LR [194] and is intended to have a range of 8,690 nm (16,090 km) [195], but with the two-class seating capacity of the 777-300ER [194], [195]. It will be a shorter derivative of the 777-9 [157], which will be produced first and which itself will be a derivative of the 777-300ER.

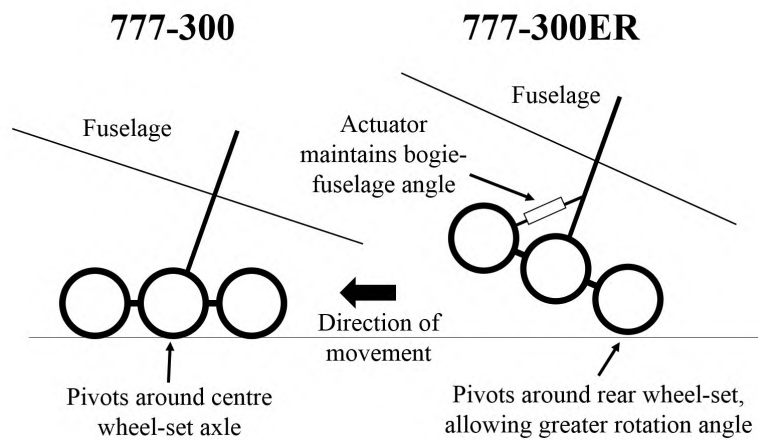


Figure 3.8: Boeing 777-300/-300ER main undercarriage at rotation during take-off.



Figure 3.9: The 777X compared with the 777-300ER (in black outline) (Source: Boeing [157]).

The 777-9 and 777-8 are expected to have fuel-burn improvements of 20% and 13% per seat, respectively, as compared with the -300ER [196]. Changes to the previous 777 airframe are reported to be as follows:

- Tailplane tip extensions of 1.52 m are to be added on each side [197].
- The vertical tail will be extended by 0.91 m [190].
- The wing will be new (but based on the 787 wing), incorporate extensive use of composites, and folding wingtips will be introduced to ensure the aircraft fit in existing airport taxiway and gates [157], [190].
- New main landing gear wheels and tires will be incorporated [197].

Table 3.11: Major variants of the Boeing 787 family.

Variant	First flight	Description
787-8	15 Dec 2009 [198]	Basic original variant.
787-9	17 Sept 2013	Stretched derivative of the 787-8 (plugs: 3.05 m fore, 3.05 m aft [199]), with longer range. Other than the fuselage, the other overall dimensions are the same as the 787-8. Significant internal changes to the airframe (see text).
787-10	31 Mar 2017	Stretched derivative of the 787-9 (plugs: 3.05 m fore, 2.43 m aft [199]). Some local airframe strengthening.

- A new strut and nacelle are developed to accommodate the new engines [197].
- Aerodynamic advances will include hybrid laminar flow control on the vertical stabiliser and natural laminar flow nacelles [190].
- The wheelbase is made longer by 1.09 m and the engines centrelines have been moved outboard by 1.1 m [190]. Presumably, this was done to make room for longer undercarriage and the large new engines.
- Compared with the 777-200, the 777-8 will have five fuselage frames added in front of and behind the wing to increase the overall length to 69.8 m [190].
- Compared with the 777-300, the 777-9 will have two fuselage frames added in front and behind the wing to increase the overall length to 76.73 m [190].

3.2.9 The 787 Family

In April 2004, Boeing launched the new, highly efficient wide-bodied twin-jet, called the 787 ‘Dreamliner’. It performed its maiden flight on the 15th of December 2009 [198]. There are currently three production variants: the 787-8, 787-9, and 787-10. These are summarised in Table 3.11 and discussed in detail below.

The first Dreamliner was the 787-8. A stretched derivative of the 787-8, the -9, was part of the family plan from the beginning [199]. According to Boeing, as quoted by [199], the main differences from the 787-8 are as follows:

- The main undercarriage trucks are larger and the undercarriage features larger tires and brakes.
- The horizontal stabiliser design was “simplified from a three- to two-piece construction”.
- A hybrid laminar-flow control system was added to the leading edge of the fin and horizontal stabiliser.
- Five fuselage frames (3.05 m) were added in front of the wing and five frames (also 3.05 m) aft of it.

- The fuselage structure was strengthened locally, by increasing gauge thickness, whereas the frame design was updated to improve weight efficiency [199]. The manner in which the frames are constructed was also changed. In some cases, such as with the nose assembly, the gauge was actually made thinner. This was because it became clear that it was overdesigned originally.
- The thrust was increased to 74,000 lb.
- The wing has the same planform and area as the -8. However, to strengthen the wing for higher operating weights, it was locally optimised to save weight. The thickness of the wing skin has been increased and some of the fittings were made larger.

Unlike the more-often employed strategy of increasing the capacity at the expense of range, with the 787-9 both were increased significantly by increasing the MTOW [199]. This was made possible by substantial weight reduction (through applying experience acquired with the earlier -8) and the use of the novel hybrid laminar flow control system [199]. These improvements had a negative impact on commonality with the -8 though. However, some of the advances obtained through the development of the -9 could be incorporated in the later -8 production aircraft [199].

The 787-10 was not actually originally conceived as part of the 787 family [200]. However, compared with the relatively large jump from the 787-8 to the 787-9, the -10 could be considered a ‘simple’ stretch. This is because it was certified with the same MTOW as the -9 [198]. This reduced the extent to which the airframe had to be strengthened [201]. About 95% of the -9 part numbers are reused on the -10, enabling exceptionally high commonality across the two variants [201].

3.3 McDonnell Douglas

McDonnell Douglas (and especially the preceding Douglas Aircraft Company) produced many famous and celebrated transport aircraft. Unfortunately, the company experienced increasing financial problems and merged with Boeing in the 1990s.

3.3.1 The DC-8

In June 1955, almost a year after the Boeing 367-80 first flew, Douglas decided to also pursue a jet transport aircraft [152, p. 116]. The aircraft would be known as the DC-8 and the design that emerged looked very similar to the 367-80 and the subsequent 707 [152, p. 116]. It first flew on the 30th of May 1958, powered by Pratt & Whitney JT3C engines [152, p. 116].

Table 3.12: Major variants of the Douglas DC-8 family.

Variant	First flight [152]	Description
DC-8-10	30 May 1958	Basic original version.
DC-8-20	29 Nov 1958	Variant of -10 with more powerful engines.
DC-8-30	21 Feb 1959	First Intercontinental version. Increased fuel capacity and operating weights, along with changes to the wing from Srs 32 onwards (see text).
DC-8-40	23 July 1959	Similar to -30 but fitted with Rolls-Royce engines.
DC-8-50	20 Dec 1960	First DC-8 variant to be fitted with turbofan engines.
DC-8-55	29 Oct 1962	Similar to Srs 50, but with increased passenger capacity.
DC-8-61	14 Mar 1967	First stretched variant. Fuselage stretched by 11.18 m [152, p. 117]. Some local strengthening of the airframe.
DC-8-62	29 Aug 1966	Extra-long-range version. Stretch of 2.03 m of Srs 55 fuselage. New wingtips, along with other airframe changes.
DC-8-63	10 Apr 1967	Same as the Srs 61, but incorporating the advances made with the Srs 62.

There were several derivatives of the DC-8 (See Table 3.12). These were referred to by McDonnell Douglas as ‘Series’ and designated by ‘DC-8’, followed by a dash (–) or the abbreviation ‘Srs’, followed by the series number. Series 10 to 50 employed the same fuselage [152, p. 117] and had identical dimensions for all other components (apart from the wings, which received small tip extensions). From Srs 60, stretched versions started to appear [152, p. 117].

Srs 10 (DC-8-10) was the original ‘domestic’ model and entered service in September 1959 [152, p. 117]. The DC-8-20 operated at lower weights than the Srs 10, but used the more powerful engines to provide improved field performance [152, p. 116].

Developed for intercontinental services [71, p. 516], the DC-8-30 featured longer ranges than the -10 and -20 and was powered by JT4A engines [152, p. 116]. The fuel capacity was increased over that of the Srs 20, increasing the range [71, p. 516]. Compared with the Srs 10 and 20, the Srs 30 had a higher MTOW, which meant that it required longer runways for take-off [71, p. 517]. Therefore, although the subseries Srs 31 still had the same wing configuration as the DC-8-21, leading-edge droop flaps were added to the wings of the -32 and -33 [71, p. 517]. In addition, the leading edge on these subvariants was modified (recontoured) to increase the

wing chord by 4% [152, p. 116]. The Srs 40 was almost identical to the Srs 30 but was fitted with Rolls-Royce Conway engines [152, p. 16]].

The DC-8-50 was the first to sport new turbofan engines, which were housed in new nacelles [71, p. 518]. For the subseries, Srs 55, the aft bulkhead was moved rearwards, which made room for an increased capacity of up to 189 passengers [71, p. 519].

According to Francillon [71, p. 520], when the DC-8 was designed, a substantial margin for growth was provided. This was done by incorporating long undercarriage struts and a pronounced upward sweep of the rear fuselage, which would ensure that the fuselage could be easily stretched while leaving enough room to still properly rotate at take-off [p. 521]. However, it was only with the Series 60 that these margins were finally exploited. Three new versions (subseries of the Srs 60) were launched in April 1965 [152, p. 117]. The Srs 61 and Srs 63 featured the same fuselage stretch of 11.18 m, whereas the Srs 62 had a shorter increase in length of 2.03 m [152, p. 117].

The Srs 61 had the same wing, tail surfaces, engines, and fuel capacity as the -55, although some local strengthening of the structure was applied and the flaps were improved [71, p. 521]. The -61 could therefore be considered as a ‘simple stretch’. The -61 had the same MTOW as the Srs 55 and, because it had a much higher passenger capacity than that model, range diminished.

The -62 was developed for extra-long ranges with the Srs 55 as baseline [71, p. 522]. Aerodynamic improvements were also implemented, which included a wingspan increase of 1.83 m with more efficient wingtips [152, p. 117]. The Srs 62 also featured higher fuel capacity [71, p. 522].

The -63 combined the longer fuselage of the -61 with the wing and powerplant advances of the -62 [71, p. 524], and was aimed at the transatlantic market.

From 1979, a company called Cammacorp converted existing DC-8-60 aircraft to be powered by the more modern, quiet, and efficient CFM56 turbofan engines [152, p. 117]. These received the Srs 71, 72, and 73 designations and were retrofits. They are not covered in further detail here.

3.3.2 The DC-9 and descendant families

In the early fifties, the Douglas company performed studies for a small jetliner with medium range to complement the DC-8, which was just launched [152, p. 144]. This aircraft was originally planned to be a reduced scale version of the DC-8 with four engines, but this was later abandoned [71]. Instead, a new design was

Table 3.13: Major variants of the McDonnell Douglas DC-9 family.

Variant	First flight [202]	Description
DC-9-10	25 Feb 1965	Original version of the DC-9.
DC-9-20	18 Sep 1968	More powerful engines and changes to slats [71, p. 538]. Same fuselage length as Series 10 [152, p. 144].
DC-9-30	1 Aug 1966	4.57 m stretch of the -10 fuselage [71, p. 539].
DC-9-40	28 Nov 1967	1.88 m stretch of the -30 fuselage [71, p. 541].
DC-9-50	17 Dec 1974	2.44 m stretch of the -40 fuselage, same wing dimensions as on Srs 30 and 40, but more powerful engines and increased MTOW [71, p. 542].

proposed, featuring a T-tail and rear fuselage-mounted turbofan engines [71].

The DC-9 was launched on the 8th of April 1963 [152, p. 114]. Little is common with the DC-8 and the DC-9 can be considered a clean-sheet design [152, p. 114]. However, by observing the nose and cockpit geometry one can notice that it is based heavily on that of the DC-8.

The DC-9 was planned for significant growth from the start of the programme [71]. In fact, it seemed more amenable to growth than the British Aircraft Corporation (BAC) 1-11, which was limited by its lower thrust engines [71]. This enabled the DC-9 to be more competitive early in its career [71]. It eventually evolved into the MD-80, MD-90, and later the Boeing 717 families. In the next subsection, the original DC-9 variants are discussed, followed by the MD-80, the MD-90, and the Boeing 717.

3.3.2.1 The DC-9

All DC-9 versions were powered by variants of the Pratt & Whitney JT8D engine. With 831 built (excluding military versions and the later MD-80), the DC-9 was the most successful of Douglas Commercial aircraft [71, p. 535]. Salient information regarding the DC-9 variants is summarised in Table 3.13. The first version of the was the Series 10. It took off on its maiden flight on 25 February 1965 [71, p. 537].

The series 20 featured a 1.22 m longer wingspan than the Series 10. This modification was originally applied to the Series 30 (which preceded the Series 20) by means of adding extensions at the wingtips. Also, like the Series 30, the Series 20 employed full-span leading edge slats for improved field performance and featured more powerful engines [71, p. 538]. The fuselage length was the same as that of

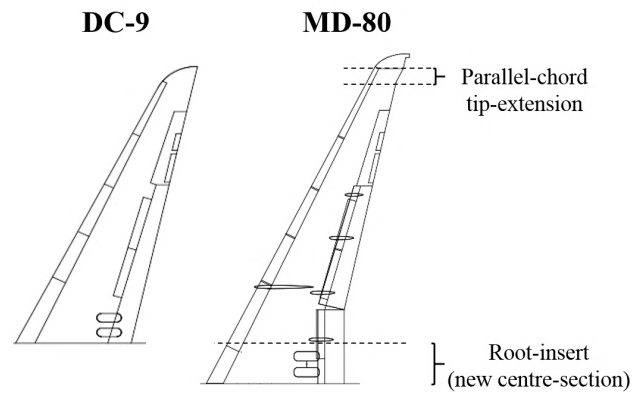


Figure 3.10: McDonnell Douglas DC-9 and MD-80 wing comparison (from Boeing CAD drawings [151]).

the -10 [152, p. 144].

The Series 30 followed the Series 10 into production (before the Series 20) and featured a 4.57 m stretch of the -10 fuselage [71, p. 539]. This was a relatively straightforward modification, because the MTOW of the Series 10 was at first limited to 80,000 lb (which was the FAA maximum restriction at the time for two-man crews) and its engines were much derated in terms of thrust [71, p. 539]. The aforementioned changes to the Series 10 wing (see the preceding paragraph) were performed to ensure that the field performance of the Series 30 remained acceptable.

The series 40 was a further stretch. The subsequent increase in payload capacity was made at the expense of range and runway length requirement [71, p. 541]. Another fuselage stretch was performed to create the Series 50, which was announced in 1973.

3.3.2.2 The MD-80

From 1975, Douglas studied how the DC-9 could be evolved further by taking advantage of the emerging turbofan JT8D-200 series of engines [152, p. 146]. As a result of these, the ‘DC-9 Series 80’ was launched in October 1977. However, it was soon renamed as the ‘MD-80’ [152, p. 146] to reflect the McDonnell and Douglas merger. It had a 4.34 m longer fuselage than the DC-9-50.

However, the wing was changed substantially (see Figure 3.10). To increase the span, a centre-section plug was inserted and parallel-chord span extensions of 0.61 m were added at the wingtips [203]. These changes allowed an increase in weight while keeping field-length performance acceptable.

Table 3.14: Major variants of the McDonnell Douglas MD-80 family.

Variant	First flight [203]	Description
MD-81	18 Oct 1979	Basic version.
MD-82	8 Jan 1981	Higher thrust engines for hot and high conditions and increased payload/range [203].
MD-83	17 Dec 1984	Extended range version. Two additional fuel tanks in cargo compartment. Higher thrust engines [203].
MD-87	4 Dec 1986	5 m shorter fuselage length than MD-81/-82/-83 and taller tailfin [152, p. 146].
MD-88	15 Aug 1987	Similar to MD-82, but with more advanced systems and increased use of composites [203].

Information on the different variants of the MD-80 are summarised in Table 3.14. At first, the MD-80 consisted of three subvariants, known as the MD-81, MD-82, and MD-83. These had the same overall dimensions, but differed in engine power, fuel capacity, and operating weights [152, p. 146]. They were followed by the MD-87, which was announced in 1985 [152, p. 146]. The MD-87 featured a shorter fuselage and the tailfin had to be extended by 25.4 cm above the horizontal [152, p. 146]. On later MD-87 models, a new low-drag tailcone was adopted [203], which became standard on subsequent variants.

In 1986, another member of the MD-80 family was launched. This variant was called the MD-88 and was similar to the MD-82 but featured a number of system refinements [152, p. 146].

3.3.2.3 The MD-90

The MD-90 represented the next generation of the family, which again attempted to exploit advances in engine technology [204]. The project was not successful and only a few variants of the MD90-30 were produced. The engines were, once again, heavier and so the fuselage had to be stretched a further 1.37 m to balance this increased weight at the back [204]. An extended range (ER) version was built, dubbed the MD90-30ER, which featured increased fuel capacity and higher weights.

3.3.2.4 The MD-95/Boeing 717

The MD-95, launched in 1995, was the next step in the development and was optimised for short flights of less than 400 nm (740 km) [205]. It first flew on the

Table 3.15: Major variants of the McDonnell Douglas DC-10 family.

Variant	First flight [207]	Description
DC-10-10	29 Aug 1970	Basic original version.
DC-10-30	21 Jun 1972	Derivative of the DC-10-10, with increased fuel capacity. The wingspan was increased slightly and a centreline gear leg was added to the main undercarriage.
DC-10-40	28 Feb 1972	Similar to the -30 but employed Pratt & Whitney engines.

2nd of September 1998 [205] and was renamed as the ‘Boeing 717’ after Boeing took over McDonnell Douglas in 1997 [206]. The fuselage was 1.45 m longer than the DC-9-30 and featured almost the same wing as the heaviest DC9s, but which was aligned at a higher incidence angle [206].

3.3.3 The DC-10 and MD-11

The DC-10 was developed by McDonnell Douglas in response to a requirement from American Airlines for an aircraft that could fly long-haul services from short runways [207]. It was a single deck widebody that employed a distinctive three-engine layout. The DC-10 was manufactured in three basic variants: the -10, -30, and -40 (see Table 3.15). All the variants had the same fuselage dimensions [157].

The Series 10 was the original, ‘domestic’ variant [207]. The Series 30 was the intercontinental version [207] and had a higher fuel capacity than the Srs 10. It was heavier and required an extra landing gear leg on the centreline of the fuselage. An extended-range version of the Srs 30, the DC-10-30ER, was also introduced and featured an extra fuel tank in the rear cargo compartment [207]. The DC-10-40 was similar to the -30ER but employed Pratt & Whitney engines [207].

The MD-11 was a major upgrade of the DC-10. It was launched on the 30th of December 1986 and first took the skies on the 10th of January 1986 [208]. The DC-10 fuselage was stretched by 5.67 m using two plugs [209]. The profiles of the wing trailing edge and tips were modified and Whitcomb-style winglets were added [210]. The tail-cone was also recontoured to reduce drag [209]. The horizontal stabiliser was made significantly smaller and featured a new advanced cambered aerofoil, revised trailing-edge camber, less sweep, and a trim fuel tank [207]. More advanced propulsion was employed. Despite these changes, much of the DC-10 tooling could be reused on the MD-11 [209].

Further stretches were planned by McDonnell Douglas [211, p. 68]. Unfortunately,

the MD-11 was not quite a success and the company experienced increasing financial problems. McDonnell Douglas merged with Boeing in 1997 and these projects were subsequently abandoned.

3.4 Airbus

The formation of Airbus reflected a rising level of European collaboration and the merging of many aerospace companies during the 1960s and 1970s. Airbus has since grown into an enormously successful enterprise, producing some of the most competitive and innovative airliners in the world.

3.4.1 The A300

Airbus's first endeavour, the A300, emerged in the late 60s to early 70s as a 252-passenger twin-engined jet transport, with a range of about 1,200 nm (2,224 km). This aeroplane was officially designated the 'A300B1' and performed its maiden flight on the 28th of October 1972 [152, p. 52].

The A300 was designed with possible future growth in payload and range capacity very much in mind. According to Gunston [212, p. 37], there was ample room in the main wing box for more fuel for longer ranges and the range was eventually increased to up to 4,400 nm (8,154 km) for the -600 model, without "any change in the external size of the wing". The geometry of the undercarriage would also not have to change if the fuselage were to be stretched to increase passenger capacity by up to 25% [p. 37]. The gross weight could be increased significantly with only minor strengthening of the airframe structure (from 137 tonnes for the B2 to 170.5 tonnes for the -600R) [p. 46].

Gunston also states that, long before the A300 even flew for the first time, derivatives were already being studied [p. 85]. These initial designs received the designations of A300B1 to A300B9. The B1 was the first aircraft to fly; the B3 was never produced, and the B2 and B4 were improvements of the B1 (as discussed below). Note that the B1, B2, and B4 (but not the later, improved A300-600) are collectively referred to in this text as the 'A300B'. An additional variant, the B10, was studied in 1973 and eventually became the Airbus A310 (see Section 3.4.2). The B9 and a later-added B11 were eventually developed into the Airbus A330 and A340 respectively (covered in Section 3.4.4). The other derivative designs were never produced.

The main A300 variants are listed in Table 3.16, along with salient information. The A300B2 superseded the A300B1 and flew for the first time on the 28th of June

Table 3.16: Major variants of the Airbus A300.

Variant	First flight [213]	Description
A300B1	28 Oct 1972	Original version.
A300B2	28 Jun 1973 [152, p. 52]	2.65 m longer fuselage than the B1. B2K featured Krueger flaps.
A300B4	26 Dec 1974	Longer range variant of B2. Strengthened airframe and additional fuel capacity.
A300-600	8 Jul 1983	Advanced version of the B4. Featured airframe changes developed for the A310.
A300-600R	9 Dec 1987	Extended range version of A300-600 with wingtip fences and trim fuel tank added in the horizontal stabiliser [213].

1973. Its fuselage was 2.65 m longer than the B1 prototypes [152, p. 52].

The A300B4 was the longer-range variant. It featured an added centre fuel tank and, for the later A300B4-200, additional fuel capacity was made available by adding fuel tanks in the rear cargo hold [152, p. 53].

A much-improved version of the B4, the A300-600 was launched in 1980 [152, p. 53]. It featured the rear fuselage geometry of the Airbus A310 (discussed in the next subsection) with the rear bulkhead moved aft, which enabled two more rows of seats to be added [212, p. 83]. The wing was almost the same as the A300B, but simpler, single-slotted flaps were adopted [212, p. 83].

Other improvements included more advanced engines, an increase in the use of composites, and simpler systems, which decreased structural weight [152, p. 53]. According to Gunston [212, p. 83], a single frame was added to increase the length of the fuselage by 520 mm. Furthermore, the smaller horizontal stabiliser of the A310 was adopted for the -600 [p. 83]. Also, to further incorporate advances made with the development of the A310, the rudder material was changed to composites and the outboard low-speed ailerons were removed [p. 83]. These modifications reduced drag by about 4%, whereas the landing lift-coefficient was increased by 8% [p. 83].

The next significant upgrade was referred to as the A300-600R. Small ‘wingtip fences’, which reduce lift-induced drag, were added, along with a trimming fuel tank in the horizontal stabiliser [152, p. 53].

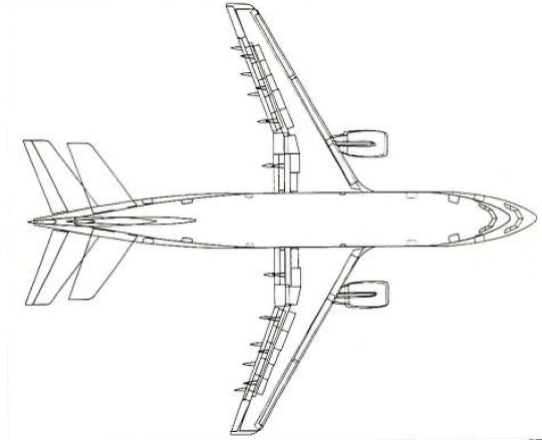


Figure 3.11: Airbus A300/A310 comparison (from Wright [214, p. 26]).

3.4.2 The A310

The first major derivative of the A300B was the A300B10 (later called the A310). A comparison between the A300 and A310 is shown in Figure 3.11.

The A310 fuselage was 13 fuselage frames shorter than that of the A300B, and the aft fuselage section was re-profiled, while the rear pressure bulkhead was moved further back to allow for more rows of seats to be added [212, p. 104]. As already stated, these rear-fuselage changes were later adopted for the Airbus A300-600.

The wing was, from an aerodynamics point of view, “completely new”, but remained structurally very similar to the wing of the A300B [152, p. 54]. Although the third inboard spar of the A300 wing was removed, the rib design and spacing remained little changed, apart from a reduction in chord to increase the wing’s aspect ratio [212, p. 100]. This had the additional effect of increasing the thickness-to-chord ratio.

According to Gunston [212, p. 100], the biggest structural modification was that the wing skins were produced in larger sections, which reduced mass at the skin joints. The outboard aileron was eliminated, as it proved to be redundant and the A300B’s flaps were replaced by new, simpler ones [p. 103]. The A310’s leading-edge slats featured larger chords and radii, which improved field-performance [214, p. 26]. Also, extensive use of newer lightweight alloys and composite materials was made [212, p. 104]. These changes to the geometry and components of the wing necessitated a redesign of the wing centre-section [214, p. 26].

The vertical tail of early A310s remained common with the A300B, whereas the horizontal tail was a scaled-down version of the A300’s [214, p. 27]. The main undercarriage was almost totally re-designed to minimise weight [212, p. 104], but the nose gear remained similar to that of the A300B.

Table 3.17: Major variants of the Airbus A310.

Variant	First flight	Description
A310-200	3 Apr 1982 [215]	Basic version. Major derivative of the Airbus A300B.
A310-300	8 July 1985 [152, p. 54]	Extended-range version. Added horizontal stabiliser trim fuel tank (later also used on the A300-600R) and optional underfloor fuel tanks [212, p. 109]. Small wingtip fences added (also implemented on later A310-200 models and used on the A300-600R [152, p. 54]). The -300's fin was made from carbon-fibre [212, p. 113].

The maiden flight of the A310 was on the 3rd of April 1983 [152, p. 54]. It was originally offered in two variants, the -100 and -200, but the -100 was later abandoned [152, p. 54]. The later A310-300 design focussed on increasing the range of the basic variant (the -200). The increase in range from the basic A310-200 was achieved with almost no external changes in geometry.

3.4.3 The A320 Family

In 1981, the decision was made by Airbus Industrie to proceed with the development of a short-to-medium range single-aisle airliner [152, p. 56]. The resulting all-new design became known as the ‘A320’. It was officially launched on the 23rd of March 1984 [216]. The wing design evolved from studies performed in Britain for the BAC Three-Eleven, which was never produced [212, p. 169].

The A320 has since developed into a highly successful family of aircraft. The members of this illustrious family are the A318, A319, A320, and A321, as well as ‘new engine options’ for these. All A320 family members share the same basic wing and tail geometry, apart from a small increase in wing area for the A321 and a higher tail for the A318.

3.4.3.1 Original A320 family members

The first flight of the A320 was on February 22, 1987 [216]. There were two versions of the original A320 – the -100 and -200, with the -200 becoming more popular[216]. From 2012, the option of having ‘sharklets’ (large winglets) was available [216]. These were also available as retrofits from 2013 on all A320 family members [216].

Table 3.18: Major variants of the Airbus A320 family.

Variant	First flight	Description
A318	15 Jan 2002 [217]	Shorter derivative of the A319 (0.79 m fuselage plug removed in front of wing, 1.6 m behind wing [217]). Taller tailfin.
A319ceo	29 Aug 1995 [218]	Shorter derivative of the A320 (1.6 m fuselage plug removed forward of wing and 2.13 m removed aft). De-rated engines. One over-wing emergency exit pair removed.
A319neo	31 Mar 2017 [218]	New-engine option of the A319. Same changes as for A320neo.
A320-100	22 Feb 1987 [216]	Original A320 family aircraft. Superseded by the A320-200.
A320-200	1987/1988 [216]	More advanced version of the A320-100. Addition of large centre-section fuel tank and wingtip fences [216].
A320neo	25 Sep 2014 [216]	New-engine option of the A320-200 (see text for airframe modifications).
A321-100	11 Mar 1993 [219]	Stretched derivative of the A320 (plugs: 4.27 m fore, 2.67 m aft) [219]. Each plug has two doors. Features more powerful engines, strengthened undercarriage, and revised trailing-edge flaps.
A321-200	12 Dec 1996 [219]	Higher MTOW variant of A320-100. Features higher thrust engines, structural strengthening, and extra fuel capacity in additional centre fuel tanks [219].
A321neo	9 Feb 2016 [219]	New-engine option of the A321-200. Same changes as for A320neo.
A321LR	31 Jan 2018 [220]	Long-range version of the A321neo. MTOW increased and a third additional supplementary fuel tank added [219]. Undercarriage and wing were reinforced [219].

The stretched A321 was launched on the 24th of November 1989 [219]. More powerful engines were fitted and the airframe and the undercarriage were strengthened [219]. The flaps were changed from being single- to double-slotted [219]. The flaps were also larger, such that it increased the chord of the wing slightly [221, p. 72] (see Figure 3.12). This was most likely done to maintain acceptable low-speed and field performance [221, p. 72]. The original A321 versions were the A321-100 and an increased MTOW, longer range/higher capacity A321-200 (launched in 1995) [219].

The next variant, the A319, featured a shortened fuselage and was launched on 10 June 1993 [218]. It was followed by the A318, the smallest of the A320 family,

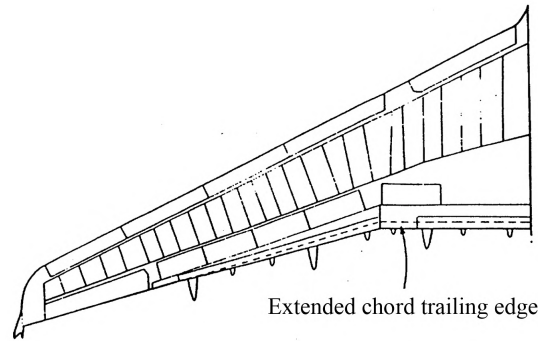


Figure 3.12: Planform view of the A321 wing showing the chord extension of the A320 baseline wing [221, p. 73].

which was launched on the 26th of April 1999 [217]. The A318 features a shortened A319 fuselage and, to ensure that directional stability remained acceptable, the vertical tail had to be made taller by means of a tip extension [217].

3.4.3.2 New Engine Option

New engine option (neo) variants of the A319, A320, and A321 were launched in December 2010 [216]. The engine options were for the CFM International LEAP-X or Pratt & Whitney PW1127G [216]. Apart from the new engines, airframe modifications include new pylons, as well as strengthened undercarriage and outer wings. All A320neos came with ‘Sharklet’ winglets [216]. A long-range version of the A321neo, the A321LR, was launched on the 13th of January 2015 [219].

3.4.4 The A330 and A340 families

The twin-jet A330 and four-engined A340 were developed in parallel, as customers could not agree whether they preferred two or four engines [222]. Both designs were, however, to employ almost identical wings, which featured large canted winglets [223]. The engine of the A330 was positioned at the same location as that of the inboard engine of the A340. This is probably an off-optimum location for a twin, but increased commonality. The A330/A340 wing changed with later variants, and some of the major modifications are shown in Figure 3.13.

The structural layout of the wing design was similar to the earlier A300, whereas, aerodynamically, the wing was “a direct relation to the A310” [225, p. 51]. The A330 was originally intended for short-to-medium range services, whereas the A340 was optimised for long-haul operations [225, p. 51]. The A340 was therefore initially much heavier than the A330 and required an extra main undercarriage bogey [225, p. 51]. The fuselage cross-section of the A330 and A340 family members is

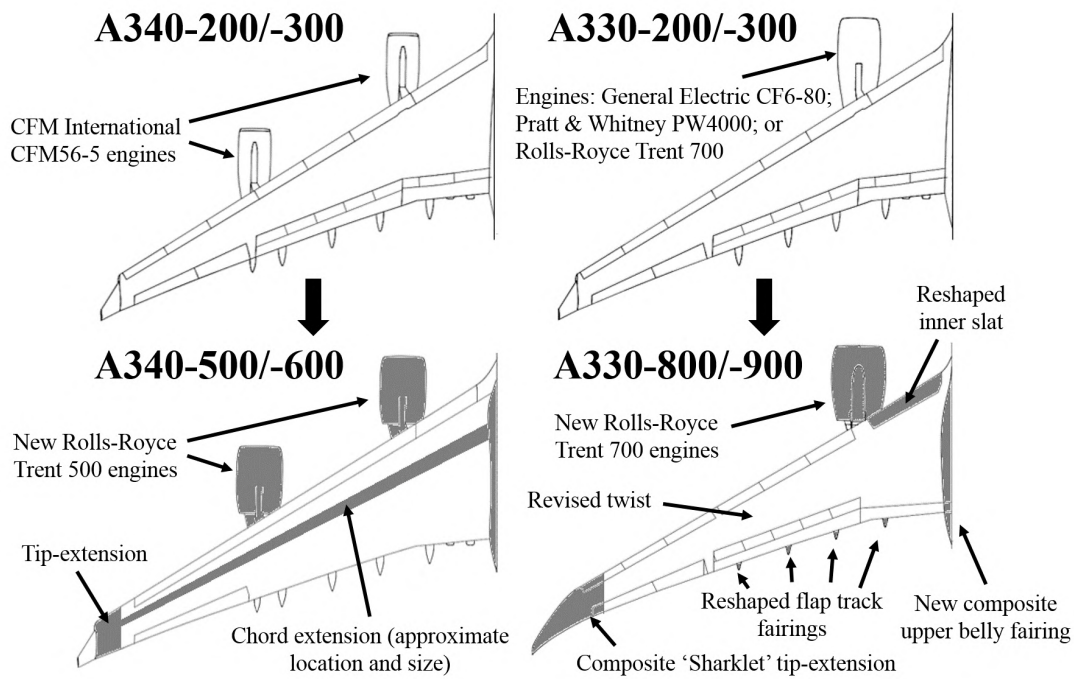


Figure 3.13: A330/A340 wing evolution (planform geometries from Airbus CAD drawings [224]).

the same as that of the A300 and A310. The fuselage construction and geometry are generally also the same as for the A300 and A310, except for the centre-section, where the new wing is attached [223]. Furthermore, the tail unit is the same as on the A300-600 and A310 (except for the A330-200, A340-500, and A340-600, which have larger tail surfaces) [223].

3.4.4.1 The A330

The A330-300 was the first version of the A330. It took off on its maiden flight on the 2nd of November 1992, which was after that of both the A340-200 and A340-300 [223], [226]. Airbus launched a shorter fuselage variant, the A330-200, in November 1995 [223]. This variant had longer range, but at a reduced payload capacity [223].

An optional increase in MTOW of up to 242 tonnes became available for both the -200 and -300 in November 2012 [223]. With this modification, the centre fuel tank of the A330-200 also became available on the -300 [223]. A regional version was announced in September 2013, which is a lower weight derivative of the A330-300 [223].

The re-engined A330 design, the ‘A330neo’ (‘new engine option’), was launched in 2014 [222]. The neo features more efficient Rolls-Royce Trent 7000 engines [222]

Table 3.19: Major variants of the Airbus A330 family.

Variant	First flight	Description
A330-200	13 Aug 1997 [223]	Shorter fuselage length, longer-range variant of A330-300. The centre fuel tank of the A340 was adopted, necessitating strengthening of the wing [225, p. 94]. The tip of the tailfin was extended by 1.04 m [225, p. 94] and the chord was increased. The fin was later shortened again by 50 cm [227, p.3 and p.5].
A330-300	2 Nov 1992 [223]	First A330 model.
A330 high gross-weight	12 Jan 2015 [223]	High gross-weight (MTOW of up to 242 tonnes) versions of the A330-200 and -300. Aerodynamic refinements, including reshaped inboard slats and shortened flap rack fairings, more advanced engines fitted [223].
A330 Regional	Launched Sep 2013 [223]	Lower weight variant of the A330-300.
A330-800	N.A.	New engine option, based on the A330-200 [222]. Same changes as applied on A330-900.
A330-900	19 Oct 2017 [223]	New engine option derivative of the A330-300 [222]. See text for modifications.

and Airbus claims a 14% fuel reduction per seat [228]. The larger engines are accommodated by new pylons and new composite nacelles with noise-reduction features [222]. They are mounted higher above the ground to ensure that the same amount of ground clearance as with the earlier A330 is maintained [229].

The first neo to fly was the A330-900. It has the same fuselage dimensions as the A330-300, but features several aerodynamic improvements [222]. The winglets are replaced by composite blended ‘sharklets’ [223], which increase the span by 3.7 m to 64 m [223]. The twist of the wing has been revised and the upper belly fairing has been replaced with a new composite fairing [223]. The flap track fairings have been reshaped [223]. Furthermore, extensive strengthening of the wing is said to have been required [229]. Despite these changes, 95% spares commonality with previous versions of the A330 is maintained [228]. The smaller A330-800 model is derived from the airframe of the A330-200 [222].

3.4.4.2 The A340

The A340 is the four-engined member of the A330/A340 family. Versions of the CFM56 (used on Boeing 737s and A320s) were adopted for the early A340s (the

Table 3.20: Major variants of the Airbus A340 family.

Variant	First flight [226]	Description
A340-200	1 Apr 1992	Shorter fuselage length, longer range variant of A340-300.
A340-300	25 Oct 1991	Original basic variant. Preceded the A330.
A340-500	11 Feb 2002	Ultra-long-range variant of the A340-600.
A340-600	23 Apr 2001	Stretched derivative of the A340-300 (see text for modifications).

-200 and -300 models) [222]. The first A340 (an A340-300) flew in October 1991 [222]. The A340-200 is the longer-range version with a shorter fuselage. Later -200 models received additional fuel tanks in the rear cargo hold, with a strengthened airframe, and more powerful engines [226].

Eyeing the 747-replacement market [225, p. 106], Airbus launched the A340-500 and -600 variants in 1997 [226]. Both were powered by four Rolls Royce Trent 500 engines [226]. The -600 was a derivative of the A340-300, featuring a 20-frame stretched fuselage [226]. Changes to the empennage included a scaled-up horizontal stabiliser, an increase in chord of the vertical stabiliser, along with a tip extension of 0.5 m [226]. This was similar to the changes made to the A330-300 fin to create the A330-200 fin. The main undercarriage had to be modified to include an extra set of wheels on the centre leg [226].

The most significant changes applied, however, were to the wing. A wedge-shaped (tapered) insert was built into the wing box to extend the chord of the wing (by 1.6 m where the wing meets the fuselage) [225, p. 108]. The fuselage also had to be extended at this location by a 1.6 m plug [226]. Furthermore, a tip-extension was introduced, which increased the overall wingspan to 63.4 m [226]. The tapered insert increased the sweep of the wing from 30° to 31.5° [225, p. 108]. The changes to the wing resulted in an 20% increase in wing area and 38% increase in fuel capacity [225, p. 108]. Increasing the size of the wing in this manner enabled many of the leading and trailing edge high-lift devices and components to remain common with the earlier A340s [225, p. 108].

The A340-500 featured the same changes than the -600 but was stretched 14 frames less (receiving a 0.53 m plug in front of the wing and a 1.07 m plug aft of it, plus the 1.6 m where the wing chord is increased). It had capacity for additional fuel in the rear centre tank [226].

Table 3.21: Major variants of the Airbus A350 family.

Variant	First flight	Description
A350-800	N.A.	Shorter variant of the -900. Not developed further yet.
A350-900	14 June 2013 [230]	Basic version.
A350-1000	24 Nov 2016 [230]	Stretched derivative of the -900 (see text for modifications).

3.4.5 The A350 Family

Airbus launched a brand-new design, called the A350XWB (eXtra Wide Body), on the 1st of December 2006 [230]. As with the 787, a conventional twin-engined layout was adopted and extensive use was made of composites in the design [230]. Three variants were planned (see Table 3.21): the A350-800, the A350-900 (to be developed first), and the A350-1000 [230]. Demand has been low for the -800 variant and it seems unlikely to be pursued. The powerplant for the A350 family was developed by Rolls-Royce and is called the ‘Trent XWB’ [230].

The A350-900 is the baseline version and first flew on the 14th of June 2013 [230]. An ultra-long range variant was launched on the 13th of October 2015 and is called the ‘A350-900ULR’ [230]. The ULR first flew on the 23rd of April 2018 [231]. The extra range is made possible by fuel-system modifications that enable additional fuel to be carried, an increase in MTOW, and aerodynamic enhancements, which include extended winglets [232].

The -1000 is a stretched version of the -900, with a six-frame plug inserted in front of the wing and a five-frame plug aft of the wing. This stretch, along with an increase in MTOW of over 30 tonnes, necessitated strengthening of the airframe, as well as a more powerful engine [233]. Compared with the -900, the wing area is increased by around 4% [230]. This was done by extending the chord of the high-lift devices and the ailerons by about 400 mm [233]. According to Jane’s [230], 90% of the wing parts were modified. Another significant change made to deal with the higher weights is the switch from four-wheel-bogies to six-wheel-bogies. To house these, the landing gear bay had to be extended by one frame [233].

3.4.6 The A380

Design work for the world’s largest passenger aircraft, the A380, commenced in June 1994. The maiden flight was on the 27th of April 2005 [234]. Thus far, only one major variant has been produced – the A380-800. Stretched and neo variants

have been considered, but, at the time of writing, these seem unlikely to be pursued further.

However, the development of an improved version, called the ‘A380-plus’, was announced in 2017 [234]. Targets for this version includes a reduction of 13% in cash operating cost and 4% in fuel consumption, as compared with the baseline -800 [235]. It will sport new large composite split winglets and upper-belly fairing improvements. The wing-camber and twist will also be modified [235].

3.5 Payload-range capability evolution strategies

In this section, the evolution strategies (or lack thereof) of the three manufacturers considered above are investigated. For this purpose, the design payload and range capabilities of the different ancestor aircraft and their descendants (or derivative) were plotted for each manufacturer in single combined payload-range diagrams. Although changes in range and capacity are not necessarily always the main drivers for evolving existing designs, they are useful in comparing how the different manufacturers respond to market demand, which is often expressed in terms of range and capacity. Furthermore, some performance aspects, such as fuel efficiency, can be ‘translated’ into range attainable with a given payload. Another reason for choosing payload and range for the comparison is that other metrics, such as operating and maintenance cost, are generally less readily obtained.

Most of the information employed for the diagrams in this section originates from the ‘aircraft characteristics for airport planning’ documents provided by the manufacturers for the different aircraft variants. The raw data, along with all references used, can be viewed in Appendix A.

The sequence of ancestor-derivatives is illustrated by means of arrows (pointing from ancestor to descendant). As will be seen, the design payload-range points of the clean-sheet designs are indicated by squares, whereas those of the derivatives are shown with filled dots. Aircraft not yet certified (at the time of writing) are indicated by empty circles. Aircraft that were developed more or less simultaneously are connected by double-pointed arrows. Where two aircraft do not have a direct ancestor-derivative relationship, but it was determined from the literature that substantial design features were indeed reused, this is indicated by grey-dashed arrows. Colours are used to indicate different aircraft families or different generations within a specific family. Dashed arrows between an ancestor and derivative indicate that major changes were made, such as new engines or a substantially revised airframe (not just the ‘simple’ addition of fuselage plugs).

A few points need to be kept in mind when interpreting these plots. These are:

- It is not always clear what the nominal design payload for a specified aircraft employed by the manufacturer was. In general, the author endeavoured to use the ‘typical passenger’ loads, if specified. Where it was known that the fuselage length remained unchanged, the passenger load of the ancestor was used, except in cases where internal space was ‘made available’ by, for example, moving the rear pressure bulkhead further aft. If the design payload mass was not known, the number of passengers was multiplied by 95 kg to estimate this value (a typical value [26, p. 142]).
- As with the payload, it is not always clear which nominal range was used during design. There are usually also several mass versions for a given type of aircraft — some of which were added after original certification. Therefore, the maximum values for Maximum Take-off Weight (MTOW) were used when determining the range from the payload range diagrams, except in cases where it was known that a higher gross weight variant was added at a specific time (which is often the case for ‘extended range’ variants).
- Should a grey dashed arrow not exist between two points in the plots, it should not be interpreted that there is no design-reuse between these two aircraft. They only indicate that a clear description of significant design reuse was obtained from the literature for the two variants concerned. In reality, no ‘new’ aircraft is truly a clean-sheet development and the manufacturers build on decades of experience, design tools, and manufacturing tooling obtained from previous projects.
- The sequence indicated are based on the certification date for each model. If this was not available, the entry-into-service date was used, which is usually a few days after certification.
- The information in the plots are not exhaustive and not every single variant is covered. The author did, however, endeavour to cover the main variants of each type.
- The observations noted from the plots are provided in line with the descriptions obtained from the literature on the aircraft surveyed, as discussed in the previous sections.

3.5.1 Boeing

Two payload-range evolution plots for Boeing aircraft are provided in Figures 3.14 and 3.15. The smaller capacity aircraft are covered in Figure 3.14 and several observations can be made from this plot, as follows:

- The swift increases in payload and range for the 707-120 and 737-100 (leading

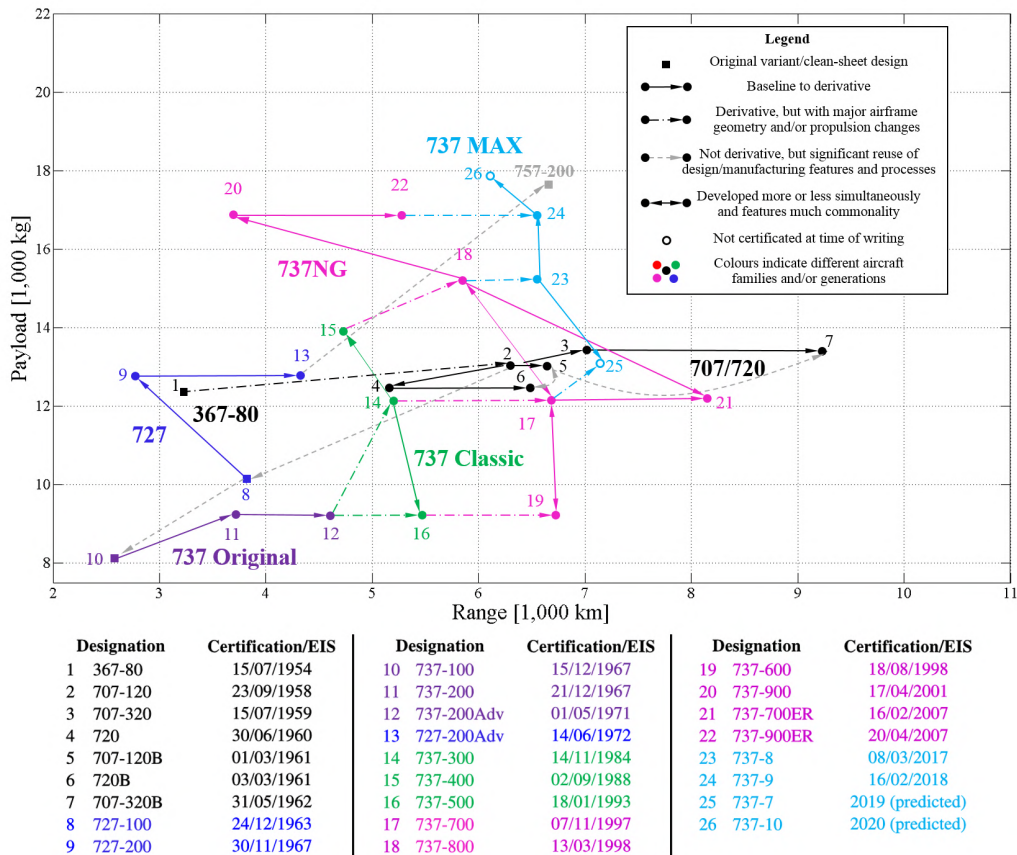


Figure 3.14: Boeing small-to-medium aircraft payload-range evolution.

to the 707-320 and 737-200, respectively) reflect the fact that these metrics were somewhat inadequate when these aircraft were first introduced. As discussed in Section 3.2, even during their respective development periods were the designs of these aircraft adapted to better match market requirements. For both aircraft this included stretching of the fuselage. For the 707-120, it also involved a substantially revised wing, whereas the 737 wing proved more amenable to change. The 727 did not need such a revision and had only undergone a ‘simple’ stretch, followed by an increase in range. Therefore, given the ease of their modifications, it therefore seems that the 727 and 737 were designed with growth being considered more than was the case with the Dash-80/707.

- The subsequent evolution of the 737 (from the 737-200Adv onwards) follows a pattern where, for each new generation, there is a substantial increase in payload and/or range (e.g. 12 to 14), followed by a stretch (e.g. 14 to 15) and a ‘virtual’ shrink of the fuselage (e.g. 14 to 16). The word virtual is used, because it is really only a reuse of a previous generation fuselage, but with the airframe modifications of the first member of the new generation, which

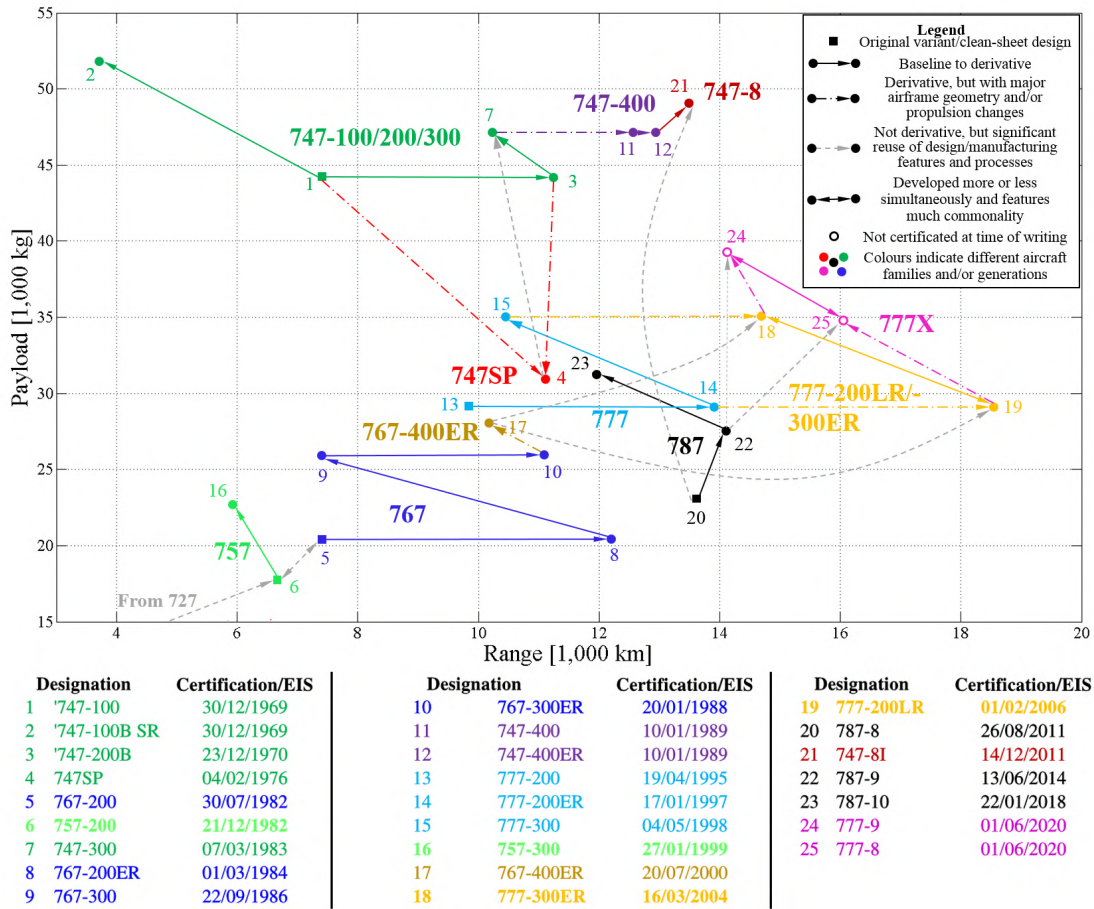


Figure 3.15: Boeing medium-to-large aircraft payload-range evolution.

usually has a longer fuselage. The author will refer to this pattern as ‘leap and branch’. It is debatable whether the 737 family was planned as such. The leap involves a substantial change in wing geometry; structures; fuel capacity; new, or more advanced versions of the engines; and a potential stretch. The ‘up-branch’ involves a relatively simple stretch, where generally the same airframe geometry as the leap is used with local structural reinforcement. The down-branch features reuse of a previous generation fuselage length, but with the new airframe modifications.

- There is a general increase in both payload capacity and range over time, and the smaller members ‘die’ out eventually.
- Larger capacity variants are added with each generation, which follows increased market demand for air travel with time.

The patterns are somewhat different with the larger capacity Boeing aircraft, as shown in Figure 3.15:

- The original 747 indeed at first experiences a ‘leap and branch’, but here the leap is the original design. The up-branch is not a stretch, but simply an

increase in payload at the expense of range. The down-branch represents the significant airframe changes of the 747SP.

- From the original 747 onwards, a distinguishable ‘Z’ pattern emerges. In this pattern, the range of the baseline is first extended through the addition of fuel capacity, higher gross weights and structural strengthening, followed by a stretch where the increased range is traded for capacity. The author is aware of at least two sources that refer to this pattern (see Refs [236] and [237]) and it appears that it is Boeing’s main strategy for systematically increasing payload and range on their long-haul products. As can be seen, there are a few exceptions, such as the 787, where the 787-9 was significantly modified, as well as the 747-8 and 777X which represent leaps. The documented extensive reuse of 787 design features (as shown by the grey arrows) suggests that significant lessons were learned on the 787, which allowed for the improvements incorporated in the 787-9 and the leaps for the 747-8 and 777X.

3.5.2 McDonnell Douglas

The payload-range evolution plot for McDonnell Douglas airliners is provided in Figure 3.16. The following observations can be made:

- For its larger aircraft, McDonnell Douglas usually focused more on increasing range than capacity. The DC-8 retained the same fuselage length for a surprisingly long time, as shown by the horizontal range-extension lines. The eventual stretch of the DC-8 (from the -55 to the -61) is substantial and did not require other major airframe modifications. This illustrates how amenable it was to growth, which makes the delay in the stretch difficult to explain. The later pattern for the DC-8 matches the ‘Z’ observed with larger Boeing aircraft.
- In contrast to the DC-8, the pattern for the DC-9 consists of a number of large consecutive leaps. These followed one another in close succession and could be attributed to competition with the Boeing 737.
- With the MD-80 series, the focus was again on increasing range, rather than capacity. With the MD-90 capacity is increased again, to compete with the newer Boeing and Airbus models.
- The MD-95 (Boeing 717) is a substantial shrink, indicating that Boeing targeted the market segment below the 737 with this aircraft. As discussed before, this did not work well, and the 717 might have been a shrink too much, being unable to compete with aircraft optimised for this segment.

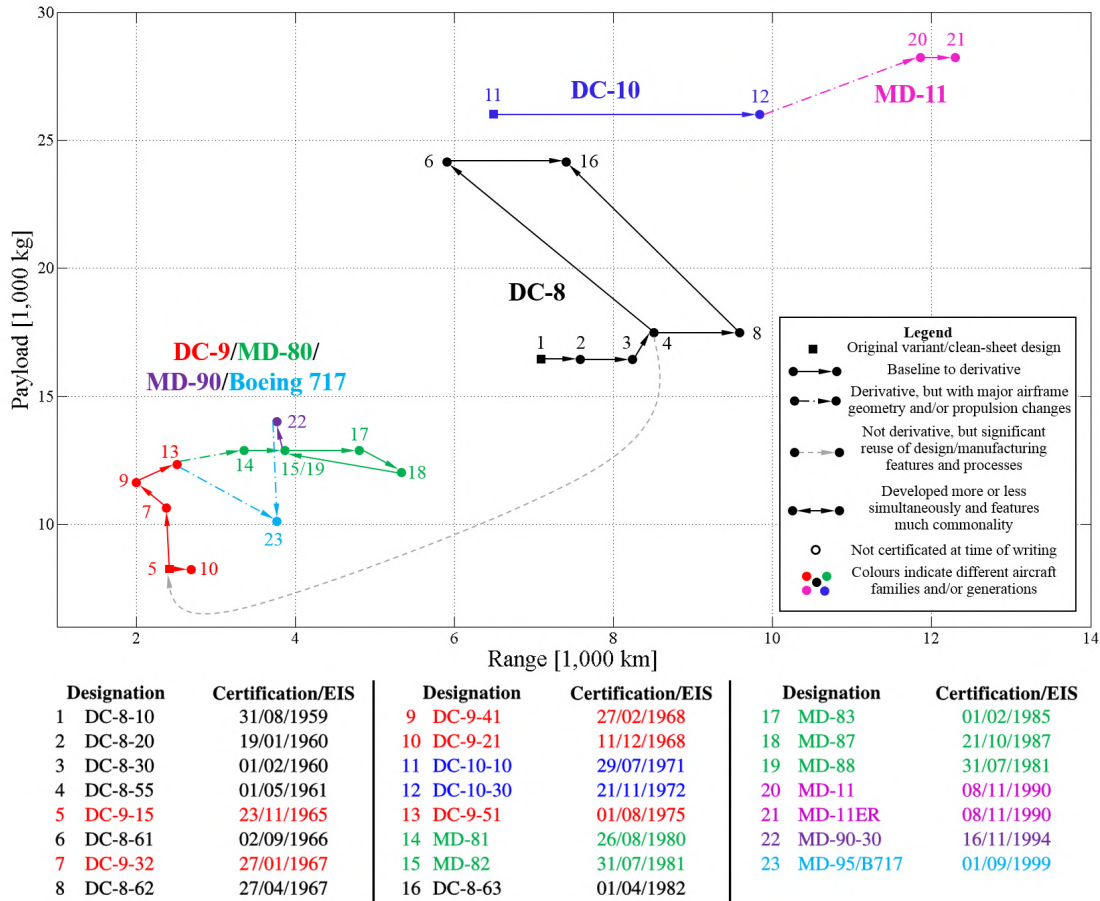


Figure 3.16: McDonnell Douglas aircraft payload-range evolution.

3.5.3 Airbus

The payload-range evolution plot for Airbus aircraft is shown in Figure 3.17. The following observations can be made:

- Airbus also generally follows the leap-and-branch method, but both for its smaller and larger aircraft. The A320 family clearly follows this pattern, but the leaps are done with fewer substantial airframe changes than with the 737. This is likely because the A320 was planned as a family from the outset and that Airbus learned from Boeing's early experiences with the 737. As with the 737, the smaller variants 'die out' with time.
- The A330/A340 is an interesting leap, as it constitutes the introduction of two different engine configurations.
- Exceptions to the leap-and-branch pattern include the A300 and A310 families, where the focus was more on increasing range and filling a niche gap in the market (for the A310). Another exception is the A350, where the changes to the A350-1000 is rather significant. This is similar to the jump

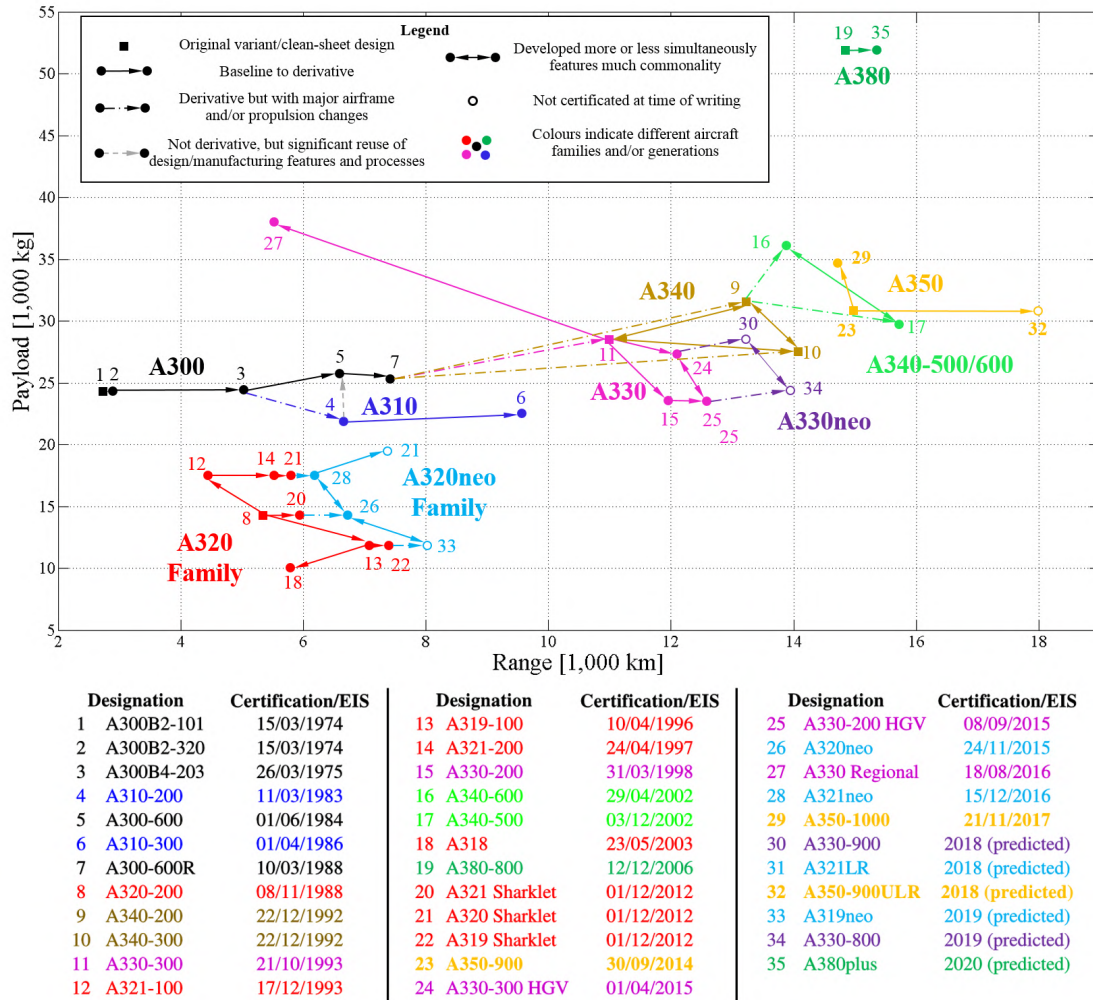


Figure 3.17: Airbus aircraft payload-range evolution.

from the 787-8 and 787-9. Finally, there is little evolution activity for the A380, which reflects the relatively low demand for this aircraft.

3.5.4 Final Notes on Payload-Range Evolution

The payload-range evolution plots (especially for Boeing and McDonnell Douglas) show a shift from somewhat ad hoc to more systematic design upgrade patterns over time. This shift is well established by the late 1970s to early 1980s. This might be because lessons have been learned and the manufacturers plan better for evolution. It might also, however, reflect the fact that there was less competition and therefore less need to adapt as aggressively as was done in the sixties. This observation is supported by the fact that Boeing and McDonnell Douglas competed fiercely in the long-range (707 vs. DC-8) and short-range (737 vs. DC-9) jet transport markets in the late sixties, and made frequent changes to their designs

in this period.

It is also interesting to note that the 787 and A350 evolution patterns differ slightly from those of their predecessors. This might be because these aircraft constituted rather substantial leaps and the experience gained enable more substantial improvements to their derivatives than aircraft that have been around for longer.

3.6 Modifications by Design Goal

In this section, the major design changes are summarised and classified according to the design goals that the manufacturers set for derivatives. For this purpose, the data collected for the payload-range evolution plots, discussed in the previous sub-section, along with all the information provided in the previous discussions on the individual aircraft families, were collated. The raw data employed in this section, along with references used, can be viewed in Appendix A.

Payload-range design goals for derivative designs can generally be classified into the following:

- Goal type 1: Increase range, keep nominal payload constant. This will usually require an increase in fuel capacity, an increase in MTOW, and structural strengthening.
- Goal type 2: Increase range and increase nominal payload. This usually involves a stretch, an increase in fuel capacity, an increase in MTOW and structural strengthening, and possibly significant airframe changes. This type of change would usually be considered a leap.
- Goal type 3: Decrease payload capacity, increase/decrease range. This usually requires shortening of the fuselage. MTOW could be kept the same, which will enable an increase in range, or it could be lowered to reduce range.
- Goal type 4: Increase payload capacity at the expense of range. This is usually done with local strengthening and inserting plugs to stretch the fuselage, while keeping the MTOW little changed. This type of change usually constitutes a simple ‘stretch’.
- Goal type 5: Increase capacity without stretching and reduce range. This is usually for regional variants of large aircraft, such as the 747-100SR and A330 Regional, which are optimised for short-range, high-density routes. Structural strengthening is usually applied to lengthen fatigue life, but the MTOW is lowered substantially.

The derivative aircraft surveyed in this study were classified into each of the above categories. The number of derivatives in each group that had a certain type of

Table 3.22: Development times for different types of derivative design goals.

Design goal	No. of aircraft surveyed	Development duration [months] ^a		
		Average	Longest	Shortest
Baseline	15	53.5	93 ^b	27 ^c
1	19	32.8	66 ^d	12 ^e
2	14	46.9	73 ^f	24 ^g
3	8	38.8	56 ^h	28 ⁱ
4	11	33.1	55 ^j	17 ^k
5	1	35	N/A	N/A

^a Duration between launch date and certification or EIS date.^b Airbus A350-900.^c Boeing 367-80.^d Boeing 737 MAX 8.^e Boeing 737-700ER.^f Boeing 747-8.^g McDonnell Douglas DC-9-51.^h Airbus A310-200.ⁱ Airbus A330-200.^j Boeing 787-10.^k Douglas DC-8-61.

change applied to it from its baseline (such as a tip extension) was then counted and reflects the frequency that the type of change was applied as one of the means to achieve the particular design goal.

Additionally, the average development time (in months) of the derivatives in each design goal category was also calculated. The development is assumed to stop at the certification date for each model. If this was not available, the entry-into-service date was used, which is usually a few days after certification. The ‘go-ahead’, launch, or first order dates were employed as the start of development. The duration, in months, between these and the certification dates was then used to obtain a rough estimate of the development time for the different aircraft. This is crude, at best, but determining the exact time a particular aircraft was conceived of for the first time is practically impossible using publicly available records. The average development times calculated only reflect data for aircraft where a launch, go-ahead, or EIS date could be found.

This development time information is provided in the following table and the change type information are shown in the images in the following subsections. In these figures, the changes applied to the baseline to create the derivative are shown in red. The number in brackets next to each change description is the number of derivative designs that exhibited that specific adaptation.

Caution should be applied when interpreting the results in Table 3.22, as only a selection of aircraft is included, and the development times are rough estimates (as discussed previously). In addition, it is difficult to determine how much resources were used for each program, which makes comparison somewhat dubious. The discussion and conclusions below should therefore be read with these caveats in mind.

However, the results match what would be expected for the different types of derivative design goals. For example, the baselines surveyed together had the highest average development time of 53.5 months, which supports the ‘common-sense’ notion that it is more difficult and expensive to develop an aircraft from scratch. The second highest average was for derivative design goal type 2 (Leaps), which reflects the difficulty associated with the more substantial changes applied to increase both range and capacity. The lowest average development times found are for types 1 (extending range, but keeping capacity the same) and 4 (simple stretches). As will be discussed below, this could be because there are relatively fewer changes to the wing and undercarriage (the derivatives that do have changed wings and undercarriage are actually more appropriately classified as leaps, but where there was only an increase in range).

As can also be seen from the table, the spread between minimum and maximum development duration are substantial. It was noted that development times have substantially increased since the sixties. More discussion regarding the data in Table 3.22 and the changes found for the associated design goals are provided in the following subsections.

3.6.1 Goal Type 1: Increase Range, keep Payload Constant

The major design changes for the goal type 1, which is a common goal type, are summarised in Figure 3.18. As can be seen, most of the changes involve providing additional fuel capacity, along with aerodynamic and structural changes to handle the associated increase in weights.

The most frequent change is the addition of centre fuel tanks. This is expected, because, to provide wing bending-relief, aircraft are usually designed with fuel tanks in the wings first [9, pp. 853-855]. As it becomes clear that more range is needed, the empty space in the centre-section of the wing becomes the obvious next location to house fuel, because of its close proximity to the centre of gravity. From there on, fuel is sometimes housed in additional tanks in the cargo-hold or in tanks integrated within the horizontal stabiliser, which can then also be used as trim tanks for longitudinal stability purposes.

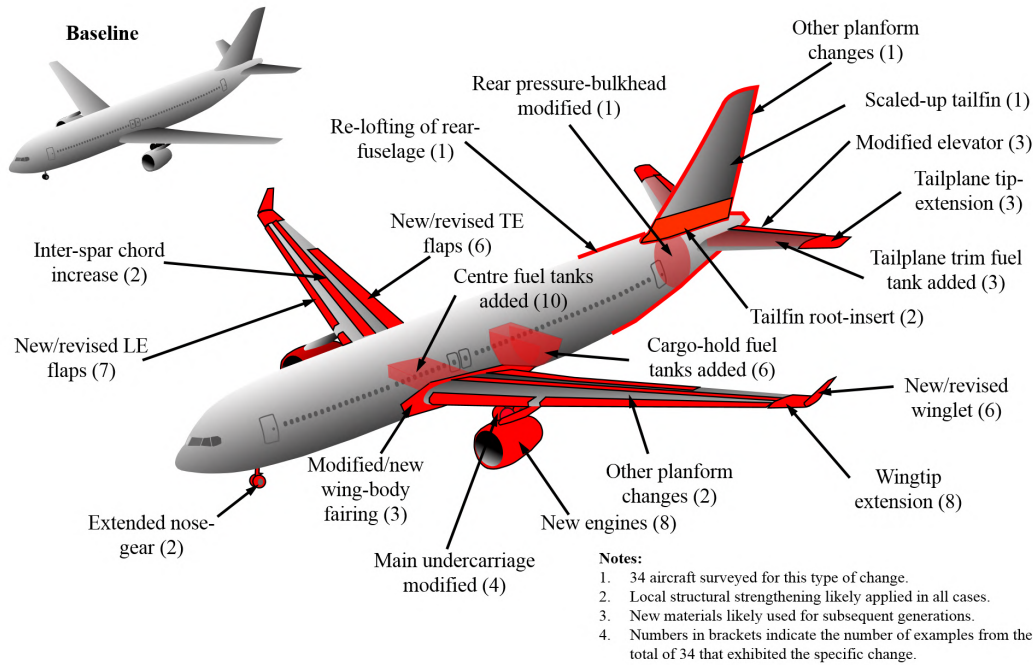


Figure 3.18: Design changes for design goal type 1.

Other frequent changes for this goal include new engines, new or revised high-lift devices, the addition of winglets, and/or the addition of cargo-hold fuel tanks. The modified flaps undoubtedly reflect the need to keep field-length performance for the higher mass derivatives within requirements. Field-length reduction would also be provided by higher thrust, which in some cases are provided by a change in the engine thrust-rating or by a brand-new engine. Also, local airframe strengthening is very often applied for this goal (a confirmation of this was found for almost every aircraft surveyed in this group), especially to the undercarriage.

Finally, the average development duration for this group was calculated to be the lowest amongst all the groups, suggesting that this is a relatively ‘easy’ goal to achieve. However, this would depend on whether the change is considered a ‘simple’ increase in range, such as with extended-range (ER) variants, or a leap. This becomes clear when considering the maximum duration found, i.e. that of the 737 MAX 8, which is comparatively very high. Indeed, the MAX 8 is the only aircraft surveyed in this group considered to be a leap, and received extensive changes, including a re-lofted rear-fuselage, new engines, and substantial modifications to the undercarriage (see Section 3.2.5.4).

to lengthen it to make room for larger engines and/or the longer fuselage to maintain clearance at rotation during take-off;

- Enlarging the empennage surfaces to maintain stability and control authority. This was usually achieved through either tip-extensions or root-inserts, and in some cases enlarged control surfaces or the scaling-up of the whole surface.

Other changes are similar to those found for goal type 1 and it was also frequently found that local structural strengthening was applied through the thickening of material gauges.

As can be seen from Table 3.22, the development times in this group are not much lower than for baseline clean sheets and, in several instances, the derivative actually took longer to develop than its baseline. Examples of cases such these are the Boeing 737-300 (with the 737-200Adv as baseline) and 747-8 (747-400ER), the McDonnell Douglas MD-90 (MD-83/MD-87) and MD-11 (DC-10-40), as well as the Airbus A340-200/-300 (A300-600R). These relatively long development times could definitely be explained by the substantial upgrades in propulsion and changes to the wing, as described above, but advances in avionics and systems across generations (not discussed here) would have contributed as well. It would, therefore, be incorrect to assume that these cases imply that there is little benefit in using a previous design as a baseline, as there are no appropriate equivalent clean sheet designs to compare them with. Additionally, the increase in development time matches an overall increase in aircraft development times across generations, as explained earlier. However, reducing effort and expenses for this derivative design goal type should be the focus of continued research, especially with regard to embedding change-capability for this goal at the conceptual design of new clean sheet aircraft designs.

There were aircraft found in this group that had comparatively shorter development periods, such as the DC-9-32 and DC-9-51. These aircraft are not quite considered leaps (as the same technology is employed as for their baselines), but also not quite ‘simple stretches’ (as the MTOW and range increased along with the increase in capacity). Additionally, they reflect the shorter development periods of the sixties.

3.6.3 Goal Type 3: Decrease Payload, increase/decrease Range

To some extent, this derivative design goal type is the complete opposite of the type explained above. However, the average development time found is the second longest of all the derivative design goal groups.

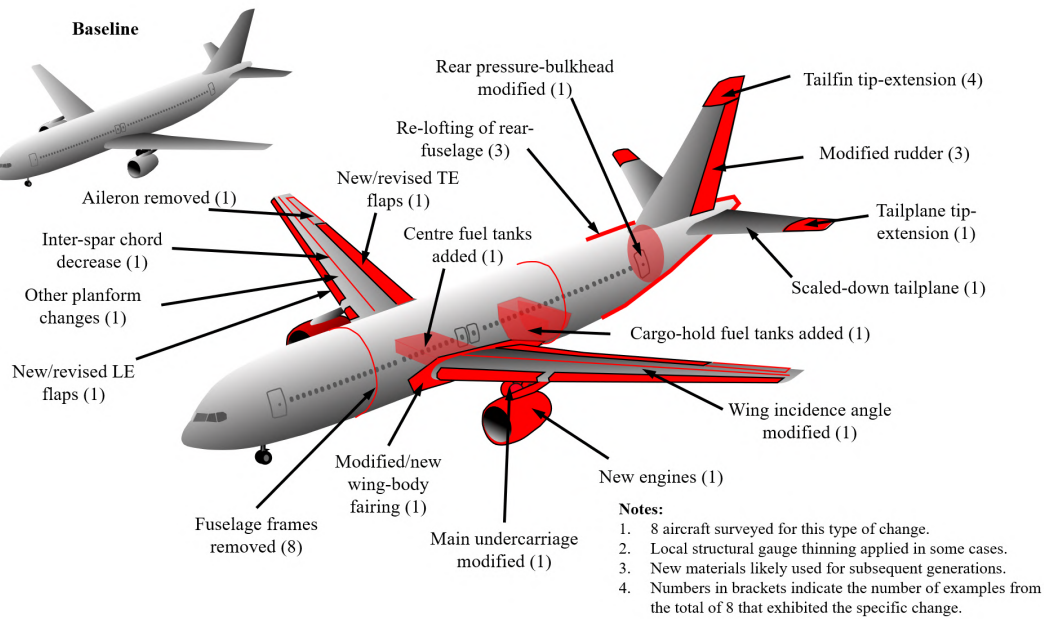


Figure 3.20: Design changes for design goal type 3.

The changes for this group are summarised in Figure 3.20. As can be seen, fuselage frames were removed in all cases to reduce the length. Re-lofting of the rear-fuselage is also relatively common – improving aerodynamics and making available additional space in the cabin (in the case of the A310). In half of the cases investigated, the tailfin span was increased by introducing a tip-extension, compensating for the reduced moment-arm associated with the shorter fuselage.

The relatively high average development time suggests that these changes are difficult to achieve. The average is somewhat biased by the relatively long development period of the A310, though, where the wing chord was made shorter, constituting a major engineering undertaking. This goal type is relatively uncommon.

3.6.4 Goal Type 4: Increase Payload at Expense of Range

This is the group in which ‘simple stretches’ are found (lengthening the fuselage with ‘minor’ other airframe and propulsion changes). As can be seen in Figure 3.21, the insertion of fuselage plugs might be accommodated by changes in flaps (to manage possible small increases in weights), winglets, the undercarriage, and new engines, but such changes are relatively rare.

From Table 3.22, it can be noted that the development periods are comparatively short for this type of goal, suggesting that it may be relatively easier to achieve than others. The longest development period found was for the 787-10, which, again, matches a general increase of average development time.

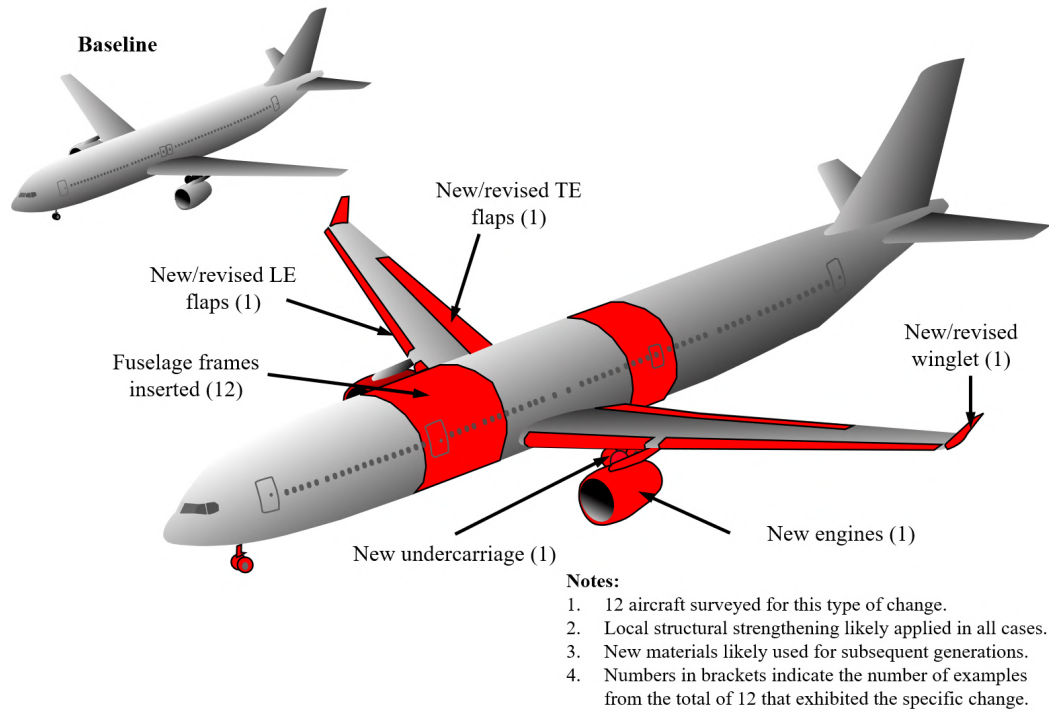


Figure 3.21: Design changes for design goal type 4.

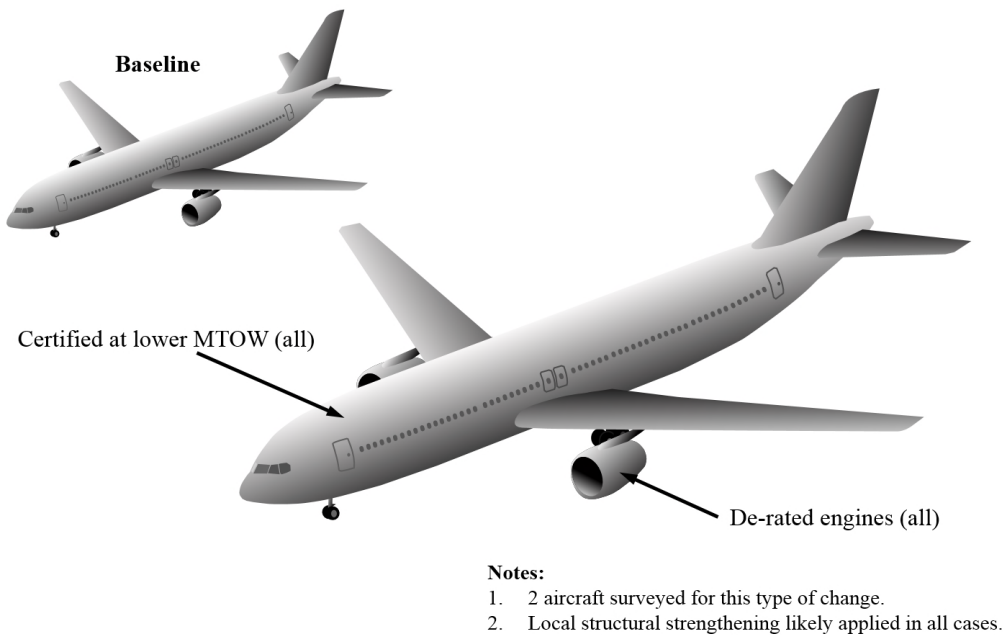


Figure 3.22: Design changes for design goal type 5.

3.6.5 Goal Type 5: Increase Payload without Stretching and Reduce Range

Goal type 5 essentially involves local strengthening of the structure, but at the same time lowering the MTOW and de-rating the engines (Figure 3.22). The purpose is to increase payload (but through filling in more seats in an unchanged cabin, rather than a stretch) and increase fatigue life for shorter, more-frequent high-capacity flights. Development duration information could only be obtained for the A330 regional.

3.7 Maximum Changes

In this section, a list of the most substantial airframe modifications identified in this chapter is provided together with the maximum increase/decrease in the geometric parameter values associated with each change. This information was employed for the work presented in Chapter 5, where the maximum changes in parameter values were used as constraints. In other words, as no published information could be found that state when a specific type of design change is impractical (as compared with a new design) the maximum values identified here were selected to be the maximum ‘reasonable changes’. The changes are summarised in Table 3.23. Note that the maximum values shown are for components which were inherited from the baseline aircraft and subsequently modified – not new components.

As can be seen in Table 3.23, some of the changes are rather large. Particularly interesting is the large modifications to chord for the Boeing 737NG, Airbus A310-200 and A340-600. Note that the percentage change in chord at the tip was calculated by using the tip chord of the aircraft with the shorter span wing and the chord at the span location of the shorter wing on the longer wing.

As will be discussed in Chapter 4, the percentage difference values in Table 3.23 were used to decide whether differences between two selected airframes may be too large for one to be considered a derivative of the other.

3.8 Conclusions

The purpose of this chapter was to obtain a practical understanding of the techniques and strategies employed by the major civil jet transport aircraft manufacturers to evolve their existing designs. To do this, the main products of three western manufacturers, namely Boeing, McDonnell Douglas, and Airbus were investigated in depth.

Table 3.23: Maximum parameter value changes for aircraft surveyed.

Description	% Change	Derivative	Baseline
Fuselage length increase	23.00%	DC-8-61	DC-8-50
Fuselage length decrease	21.50%	747SP	747-100
Wingspan increase	18.70%	737NG	737 Classic
Wingspan decrease	2.10%	A310-200	A300B4
Wing 1/4 chord sweep increase	1.5°	A340-600	A340-300
Wing chord length increase at tip	26.54%	737NG	737 Classic
Wing chord length decrease at tip	24.64%	A310-200	A300B4
Wing chord length increase at root	16.00%	A340-600	A340-300
Wing chord length decrease at root	16.16%	A310-200	A300B4
Vertical tail area increase	17.41%	737NG	737 Classic
Horizontal tail area increase	27.50%	A340-600	A340-300
Horizontal tail area decrease	7.85%	A310-200	A300B4
Main undercarriage length increase (based on track)	9.37%	737NG	737 Classic

Not surprisingly, one of the most commonly reused major airframe design features was found to be the fuselage cross-section. The fuselage length is also often changed by adding or removing fuselage frames (plugs), whereas the nose – a complex structure – is usually retained across generations. The tubular construction of most modern airliner fuselages is especially amenable to such ‘stretching’, which, amongst other reasons, explains a general reluctance amongst OEMs to diverge from this concept.

Changes to wing geometry are normally limited to only an addition of wingtip span extensions. However, in several notable cases, span-extending root-inserts, or chord-extensions, were incorporated. The Airbus A310 was the only case found where the chord was actually reduced. The leading- and trailing-edge flap type and geometry are also commonly changed, sometimes to increase wing area to improve field performance.

It was found that modifications to empennage surfaces are frequent and usually involve tip-extensions, root inserts, or photographic scaling of the whole surface.

In addition, as would be expected, changes in structural strength were also found to be very common. This is normally performed by making the material gauges thicker or thinner. Generally, however, the overall structural layout remains little

changed across generations.

Plots were created to visualise the evolution of payload and range capabilities of the different baselines and derivatives for each manufacturer. Two findings from these are worth highlighting.

- In the early jet-age (1950s and 60s), it the approach to derive derivative designs was somewhat ad hoc and large design changes were frequent.
- Later, two distinguishable patterns start to appear. The first is what the author refers to as ‘leap-and-branch’, where a ‘leap’ (a new aircraft, or a major derivative) is followed by a ‘branch’ upwards (a stretch at the expense of range) and a branch-down (a shrink, gaining range). The branches often involve re-using the fuselage and other features from the previous generation, in addition to the airframe upgrades of the leap member. Therefore, the branches usually have two ‘parents’ – the leap member and the previous generation member of similar size.) As time moves on, smaller variants ‘die out’ and a larger variant becomes the baseline for the next leap (this is noticeable in the 737 family). The second pattern is a ‘Z’ shape, where the range of the baseline is systematically increased, followed by a stretch where the range gained is traded for payload. The description of the Boeing 777 in Nield [236], confirms that the ‘Z’ pattern and branches is envisioned from program inception (at least for Boeing), but that derivative ‘leaps’ were not always planned for. Finally, the patterns undoubtedly appear because the OEMs try to meet diverse customer needs and new technology becomes available, but it also became clear throughout this study that it is driven extensively by competition (especially with derivative leaps).
- Mimicry (using similar design practices) was found to be pervasive across the three manufacturers. The mimicry between the A300 and A310 and between the Boeing 787, 737 MAX, 747-8, and 777X aircraft are particularly notable examples.

The changes made to baselines were then investigated further by grouping them according to the assumed payload-range capability goals set for the derivatives. Average development times were estimated by finding the difference in months between the launch date and the date of certification. This is a crude approximation and these conclusions should be interpreted keeping this in mind.

The first goal type was keeping the capacity the same while increasing range. This was found to be a common goal and changes are usually focussed on increasing fuel capacity by inserting centre-, under-floor cargo compartment, or tailplane fuel tanks. This is sometimes accompanied by aerodynamic, propulsion, and structural modifications to handle the increase in weights. The average development time

was relatively short for this goal type. The second goal type is to increase both payload and range. This was also found to be a common goal but involved much more extensive changes, especially stretched fuselages, large increases in wing area, changes to the undercarriage, and others. This is the group in which ‘leaps’ were found the most often and had the highest average development time. With the third goal type – decreasing payload capacity, with an increase or decrease in range – relatively few modifications were found, apart from the obvious shrink in fuselage length. The average development time was found to be rather long for this group, however. Goal type 4 involves simple stretches. Airframe modifications for this type were also usually minor and development times tended to be short. Goal type 5 is when a ‘regional’ variant is produced. This involves decreasing maximum take-off weight while increasing strength to increase fatigue life for short-haul, high capacity operations.

Finally, maximum changes found for salient geometrical design parameters were documented. This information was used as design constraints in the work presented in Chapter 4.

CHAPTER 4

Automatic Airframe Similarity Assessment

Chapter Summary: In this chapter, a computational framework for automatic airframe similarity assessment is proposed. It is intended for predicting similarity, which could be used as an indication of possible re-design effort between baseline and descendant airframes that may have substantially different configurations. The framework consists of a suite of tools that enable airframe partitioning and commonality prediction. The airframe partitioning exploits design-structure matrices and an object model to facilitate automatic identification of component pairs that must be assessed in detail for similarity. The prediction of commonality between the airframes is made possible by several proposed algorithms, one of which solves a longest common subsequence problem to find common segments in fuselages and nacelles. For wing commonality, a simple multi-objective optimisation procedure is proposed that can identify the size of wing-tip extensions and/or root inserts for the longer wing, such that increases in chord length, spanwise length of the root-insert, and the difference in wing areas are minimised. Simple commonality prediction techniques are also proposed for the undercarriage, empennage surfaces, and other supporting components.

4.1 Introduction

In the motivation to this thesis, it was shown that significant savings in terms of computational expense could be obtained for evolvability exploration studies if the near- and far-future design spaces were to be evaluated separately before assigning commonality. It was also shown there that this would presuppose a method for assessing similarity between the designs. Furthermore, it was required that this method be automatic and that it could assess similarity across airframes having dissimilar configurations.

The work in this chapter is proposed to fulfil this need. Particularly, a framework for automatic airframe similarity assessment is proposed. This framework consists of several computational techniques that facilitate partitioning the airframes into their constituent components, determining which component pairs should be dispatched for detailed similarity detection, and identifying sub-segments in these components that could be made common, based on their similarity.

The chapter is structured as follows: first, an overview is provided in Section 4.2, in which the main elements of the framework, their significance, and their relationships are discussed. This is followed by an in-depth description of the knowledge representation employed (Section 4.3), after which the different inference mechanisms are introduced. The first of these, called DSM-based airframe component partitioning, which automatically determines which component pairs should be assessed in detail for similarity, is provided in Section 4.4. This is followed by descriptions of the commonality prediction techniques. Finally, conclusions are drawn in Section 4.6.

4.2 Overview

The framework is illustrated in Figure 4.1. It comprises a computational process that involves several sub-processes, which are shown in rectangles. The sub-processes are referred to here as ‘inference mechanisms’, or ‘inference modules’, and manipulate ‘knowledge’ (i.e. structured data), represented by the skewed rectangles in the figure. Each of the inference methods is supported by a set of computational enablers, which will be elaborated upon in the following sections. It can be inferred from this discussion that the framework essentially describes an architecture for a knowledge-based engineering (KBE) application. Note, however, that it was outside the scope of this research to develop and implement the framework formally as a fully-fledged KBE system.

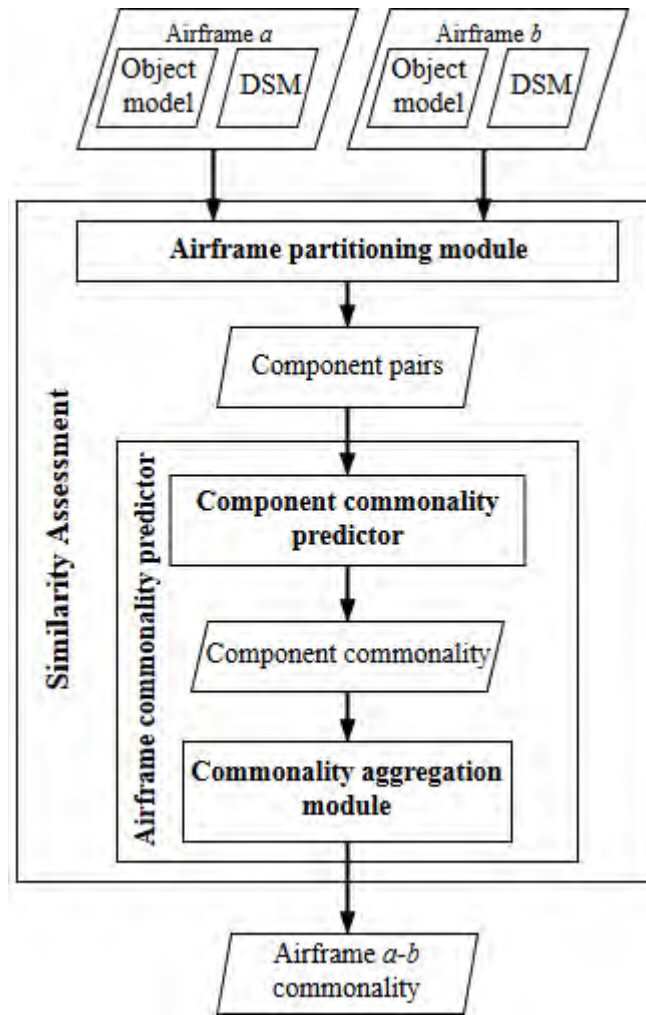


Figure 4.1: Overview of the automatic similarity assessment framework.

The process is initiated by a user, who selects two airframes (a and b) to be compared with each other. Each airframe is described by its own instance of a generic airframe object model and a design structure matrix (DSM). The object model contains structured information on the geometry, materials, as well as development and manufacturing processes associated with the different airframe components. The DSM, in turn, represents how these components are connected to each other. A generic DSM for major airframe components, which could describe most types of aeroplane airframes was developed for this research. The object model and DSM constitute the ‘knowledge representation’ of the framework and is discussed in more detail in Section 4.3.

The input airframe designs could be of existing aircraft in the manufacturer’s product portfolio (i.e. aircraft that have actually been produced); previously rejected designs that have become ‘interesting’ again; or new designs under study.

The only pre-requisite for use in the presented framework is that they are represented in accordance with the stipulated object model.

Once two airframes have been selected to be compared, their class instances and DSMs are dispatched to the ‘Airframe Partitioning’ inference module (described in detail in Section 4.4). This module interrogates the information contained in these structures to decide which component pairs across the two aircraft should be compared in detail with each other. In other words, it will break down the airframes into their constituent components and send pairs of components (i.e. one from each aircraft) of the same type to be queried for similarity, if certain criteria are adhered to. For example, it will decide whether the vertical tail of Airframe *a* can be compared with that of Airframe *b* and, if so, it will send class instances describing these to the other inference modules for detailed comparison. Note that the Airframe Partitioning module can also handle multiple components of similar type. For example, if Airframe *a* has one pair of nacelles (i.e. two engines), but Airframe *b* has four, it would determine whether the nacelle-pair of Airframe *a* should be compared with both pairs in Airframe *b*, just one, or none of them, depending on the salient characteristics.

As shown in Figure 4.1, the component pairs are sent to the ‘airframe commonality predictor’. This module compares the individual pairs of components and aggregates the results back to aircraft level, in order to provide an estimate for the overall degree of similarity between the two airframes. To assess similarity, the airframe commonality predictor estimates the potential mass commonality (which could be weighted by, for example, cost) across the two airframes. It performs a detailed calculation on each of the components and produces a final airframe commonality score. This score would be between 0 (for no commonality, i.e. no similarity) and 1 (for full-commonality, meaning full reuse of the baseline design).

Each of the elements of the framework discussed above will next be elaborated upon in more detail.

4.3 Knowledge Representation

As stated before, the knowledge representation aspect of the proposed framework comprises the object model and DSM of the airframe. These are discussed separately in the subsections that follow.

4.3.1 Aircraft Object Model

An object model for a generic aircraft, which is composed of the airframe (of interest here), as well as the engines and systems, was developed, as shown in the Unified Modelling Language (UML) class diagram description in Figure 4.2. In the text that follows, the class and attribute names will be printed in *italics*.

The airframe model is represented by the white blocks in the figure and the relationships between them. As can be seen in the figure, all the constituent airframe parts are types of airframe components (i.e. of class *AirframeComponent*), which, in turn, are types of aircraft components (of class *AircraftComponent*). The expanded *AircraftComponent* and *AirframeComponent* classes are shown in the UML model in Figure 4.3. Note that, to save space in the discussions on the classes that follow, only the most relevant attributes are shown. The methods associated with the classes are not displayed either, as these are only pertinent to geometry generation and not to similarity assessment.

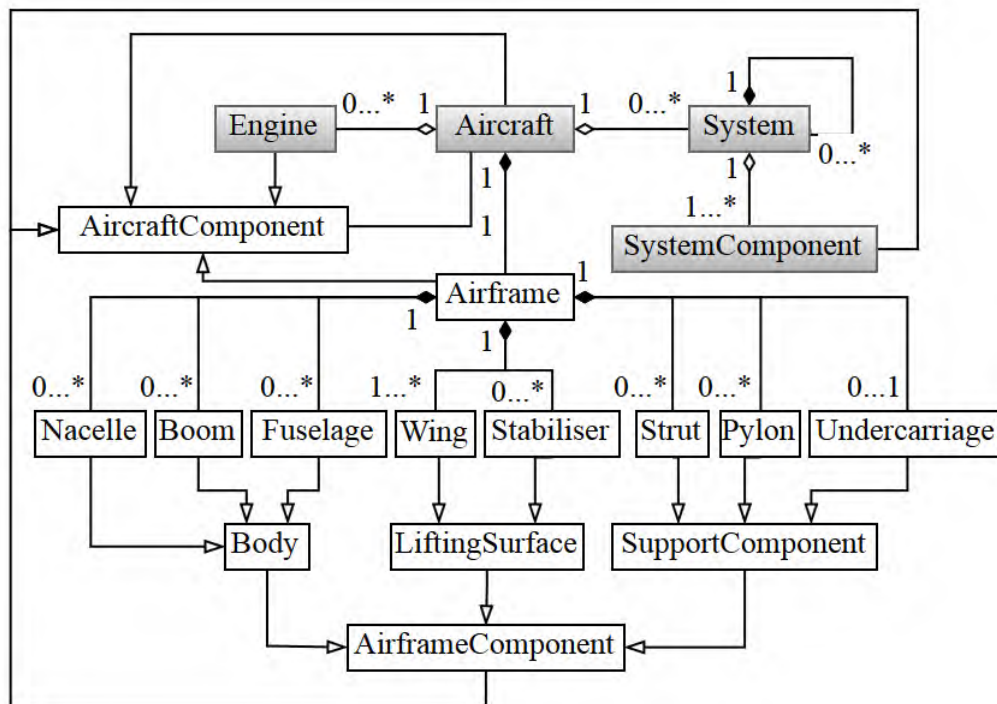


Figure 4.2: UML class diagram for the aircraft object model.

Descriptions of the most relevant attributes are provided in Table 4.1, for the *AircraftComponent* class, and in Table 4.2, for the *AirframeComponent* class.

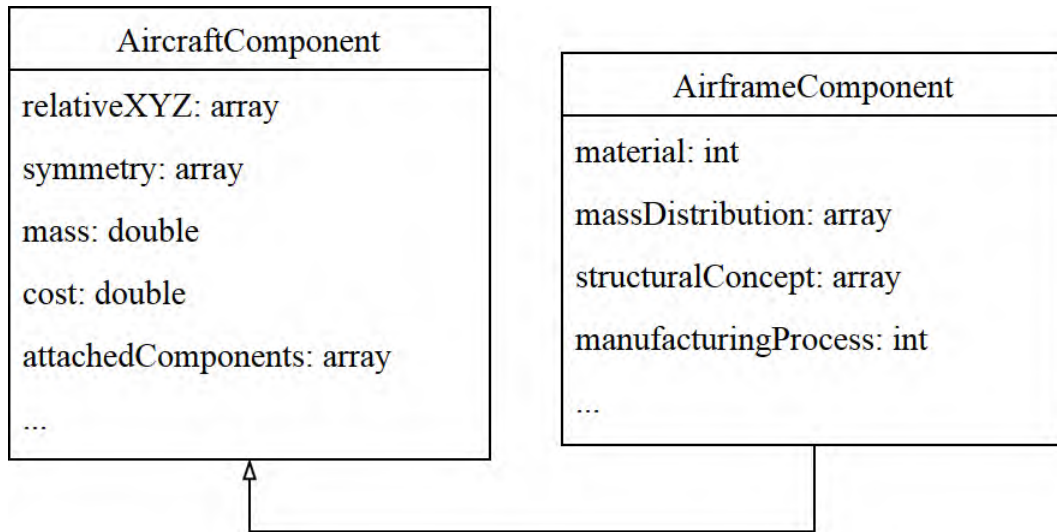


Figure 4.3: UML description of the *AircraftComponent* and *AirframeComponent* classes.

Table 4.1: Description of salient *AircraftComponent* attributes.

Attribute name	Description
relativeXYZ	Stores the three-dimensional spatial coordinates (relative to the global origin) of a reference location that is used for geometry generation purposes.
symmetry	Lists the axes around which there is symmetry (if any), as well as what type of symmetry it is (e.g. reflective).
attachedComponents	Array that lists all the identification numbers of other components attached to this specific aircraft component (in the first column), along with the type of connection (in the second column). An example of a type of connection could be, for instance for a fuselage-wing joint, a box carry-through structure, ring frames; a bending beam, or a strut-braced attachment (see Raymer [9, p. 230]).

Table 4.2: Description of salient *AirframeComponent* attributes.

Attribute name	Description
material	Main structural material from which the component is manufactured.
massDistribution	Contains values of the mass per unit length at specified locations along the length of the component (streamwise [<i>x</i> -direction] for body objects and spanwise [<i>y</i> -direction] for lifting surfaces).
costDistribution	Array representing the development cost per unit length of the component.
structuralConcept	Integer identifier indicating which structural philosophy is employed (e.g. Keelson; stringers; or longerons (refer to Raymer [9, p. 229]) for a fuselage).
manufacturingProcess	Integer identifier indicating which combination of manufacturing processes is employed.

The *costDistribution* attribute of the *AirframeComponent* class is optional, but desired, because it would provide a more complete indication of the effort to produce a specific component. Its purpose in the similarity assessment techniques is to capture the additional design expense due to the novelty of a component (due to, for example, a new technology, material, or manufacturing process being employed), or even its complexity¹. Should cost information (which would, in any case, undoubtedly be a rough estimate) not be available and if the degree of novelty is high for the derivative aircraft, the mass distribution could be weighted with weighting coefficients (obtained from designer elicitation) to capture the additional effort that might be incurred.

For struts, pylons, and undercarriage, which are ‘support’ components (of class *SupportComponent*), the mass and cost arrays will contain only the total mass and cost of the component.

It should be possible to model the salient characteristics of most conceivable aircraft (aeroplanes, at least) using this object model. It even allows such exotic concepts (for civil aircraft at least), as buried engines and blended-wing-body configurations, amongst many others, to be modelled. Note that not much attention will be granted to the support components in this text, although simple similarity assessment methods will be provided. The *Body* and *LiftingSurface* classes, which

¹Complexity could be accounted in the form of additional attributes in the *AirframeComponent* class, such as, for example, number of parts, number of interfaces, or any other complexity metric.

are comparable to the fuselage and wing “High Level Primitives” introduced by La Rocca [38], are described in more detail next.

4.3.1.1 Body Class

A UML class diagram description of the body object model is provided in Figure 4.4. As can be seen, the body (class name *Body*) has two or more cross-sections (of class *BodyCrossSection*), which determines its cross-sectional shape in the longitudinal (x) direction (the cross-sections are parallel to each other, being perpendicular to the x -axis). An example of the geometry representation of an instance of the *Body* class (for a fuselage) is shown in Figure 4.5. Descriptions of the most important attributes are provided in Table 4.3, whereas other relevant attributes and their explanations are illustrated in Figure 4.5.

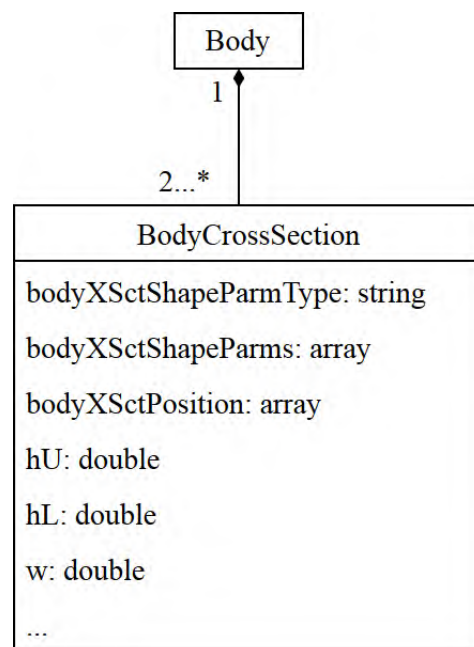


Figure 4.4: UML description of the *Body* object model.

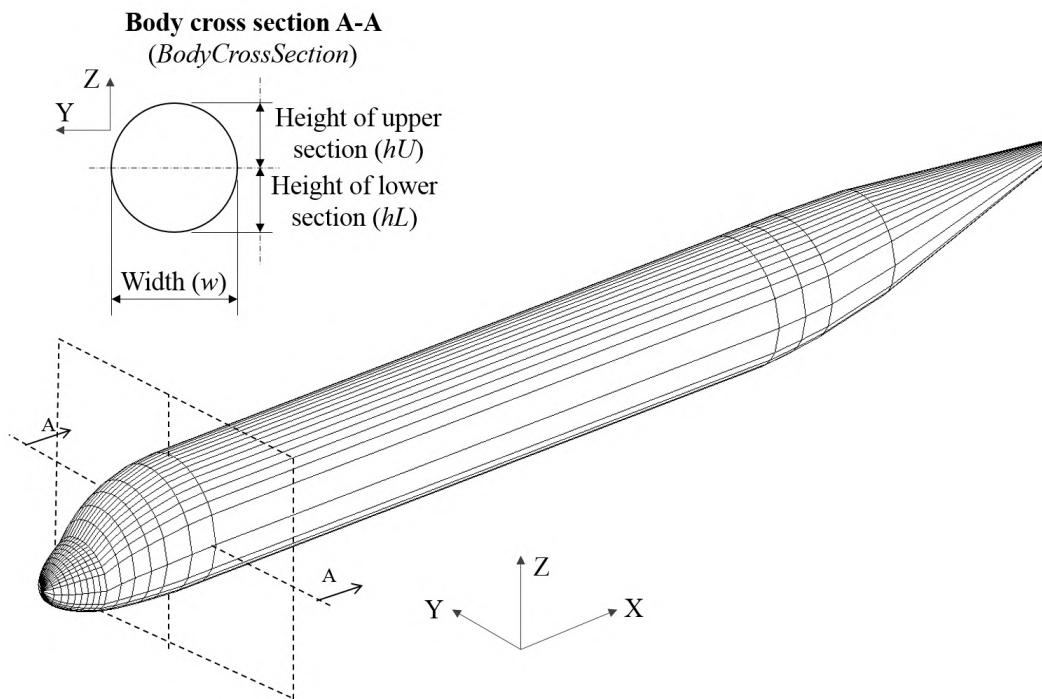


Figure 4.5: Geometry rendering of an instance of the *Body* class — in this case, a fuselage. Class and attribute names are shown in *italics*.

Table 4.3: Description of salient *Body* attributes.

Attribute name	Description
bodyXSctShapeParmType	Parameterisation technique employed to generate the shape of the cross-section. This could be, for example, the body parameterisation suggested by Kulfan [29], or the circular (elliptical), double circular, or hyper-elliptic parameterisations in the <i>PACELAB</i> [238] aircraft preliminary design software, amongst others.
bodyXSctShapeParms	Stores the cross-section parameter values.
bodyXSctPosition	Contains the coordinates of the reference point for constructing the shape of the cross-section, with respect to the origin (<i>relativeXYZ</i>) of the of the parent <i>Body</i> class.

4.3.1.2 Lifting Surface Class

The class diagram for the *LiftingSurface* class is shown in Figure 4.6 and a geometry representation of an instance of this class (for a wing) is provided in Figure 4.7. The type of lifting surface is specified with *lSurfType*, which indicates whether it

is, for example, a wing, a fixed surface, a fixed stabiliser (horizontal, vertical, or oblique) with control surfaces, or an all-movable control surface, amongst others. The lifting surface consists of one or more segments (*LsSegment*), which are used to define the planform shape and to demarcate the locations of control-surfaces. Each segment has an inner and outer lifting surface station (*LsStation*), which define the aerofoil attributes and control-surface chords.

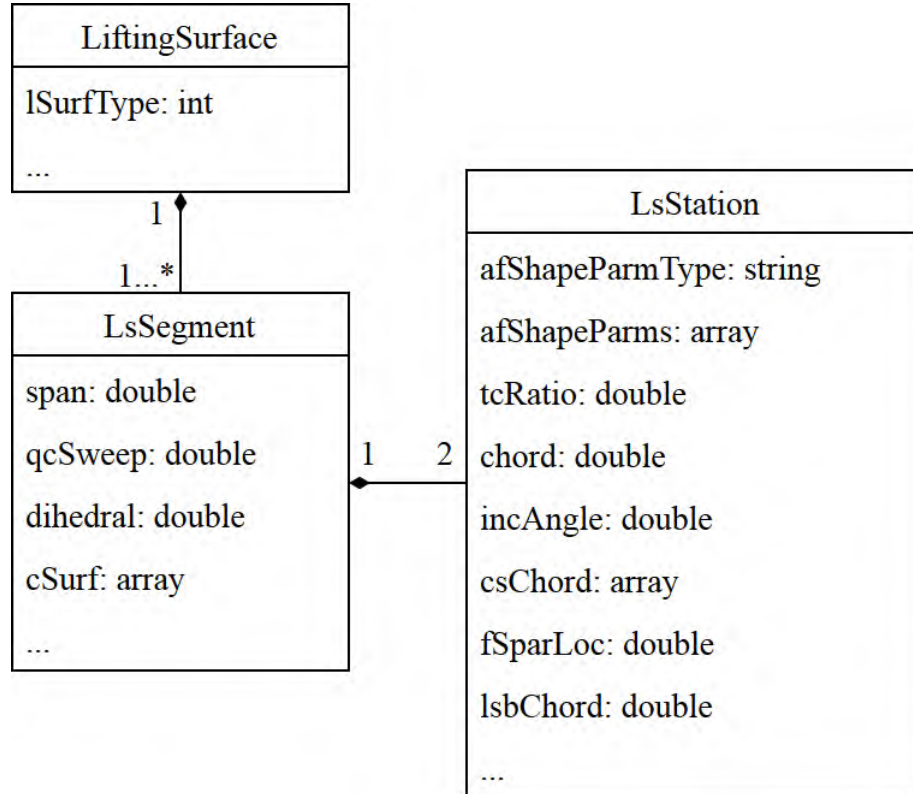


Figure 4.6: UML description of the *LiftingSurface* model.

The segments (see Figure 4.8 and Table 4.4) are ‘attached’ to each other in the order that they are listed, in a spanwise and outwards direction (positive y -direction for non-symmetrical horizontal surfaces, positive z -direction for vertical surfaces, or both positive and negative y - or z -directions for symmetrical surfaces). The lifting surface stations (see Figure 4.9 and Table 4.5) are situated at the locations where the segments meet (i.e. their relative x , y and z starting points are determined by the span [*span*], quarter-chord sweep [*qcSweep*], and dihedral [*dihedral*] of the segments). The stations could have been allowed to be defined within segments, but for simplicity in specifying control surface locations and chords, and to enable features such as a ‘dog-tooth’ to be defined, it was decided that each segment be bounded by the stations. If the user wishes to assign more stations (for example, to more accurately define the spanwise shape of the surface) he/she could do so by simply introducing more segments.

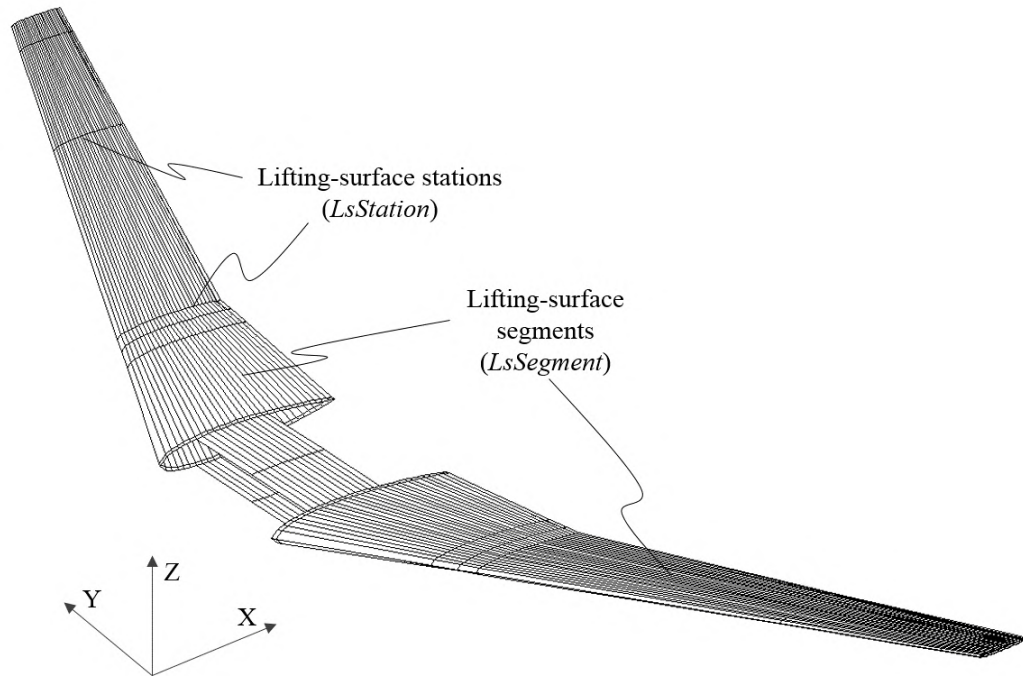


Figure 4.7: Geometry of an instance of the *LiftingSurface* class – in this case, a wing. Class names are shown in *italics*.

The *fSparLoc* and *lsbChord* attributes of the *LsStation* class enable the user to define structural members, such as the wing-box, fixed leading- and trailing-edges, as shown in Figure 4.9.

Table 4.4: Description of salient *LsSegment* class attributes.

Attribute name	Description
span	Span of segment
qcSweep	Quarter-chord sweep.
dihedral	Dihedral angle of the segment.
cSurf	An n -by-3 array, where n refers to the number of control surfaces present within the specific segment. This array lists, for each control surface, an identification number in the first column, its location (leading-edge, mid, or trailing edge) in the second column, and the type of control surface (which could be, for example, slat, single-, double-, or triple slot-flap, aileron, elevator, and so forth) in the third (optional).

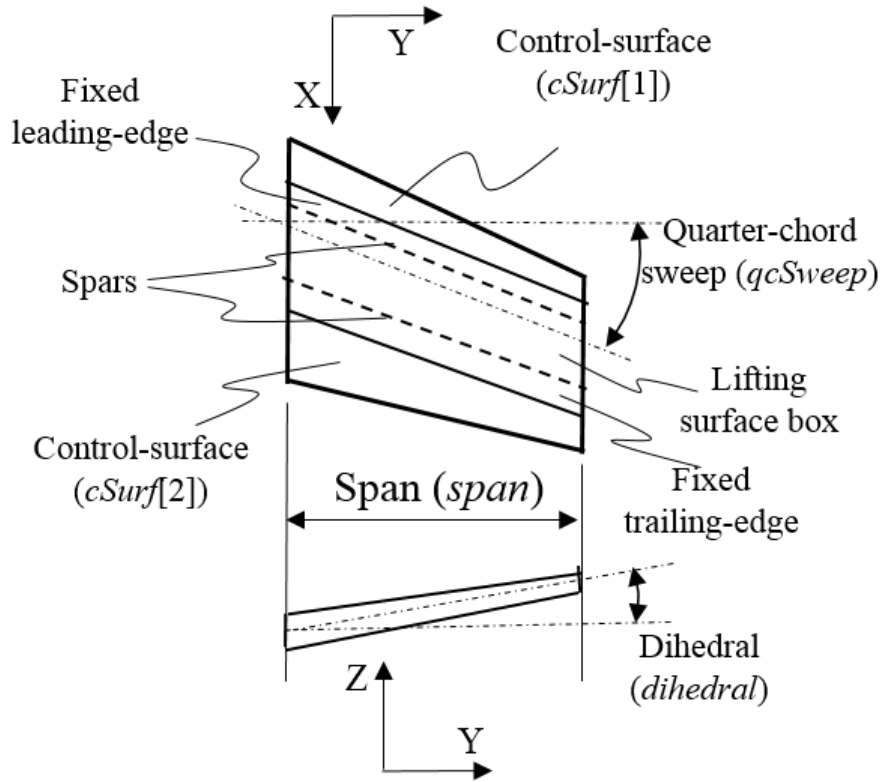


Figure 4.8: Description of the attributes of the *LsSegment* class. Attribute names are shown in *italics*.

Table 4.5: Description of salient *LsStation* class attributes.

Attribute name	Description
<i>afShapeParmType</i>	Aerofoil parameterisation type (e.g. Kulfan [29]).
<i>afShapeParms</i>	Lists the actual parameter values.
<i>tcRatio</i>	Chord length.
<i>incAngle</i>	Incidence angle.
<i>chord</i>	Thickness-to-chord ratio.
<i>fSparLoc</i>	Distance between the leading edge and the front spar (optional).
<i>lsbChord</i>	Chord of the lifting surface structural ‘box’ (specified as a fraction of <i>chord</i>) (optional).
<i>csChord</i>	An n -by-1 array listing the chord length (distance between leading and trailing edge) of the different control surfaces. The entries correspond to those in <i>cSurf</i> of the <i>LsSegment</i> class (see Table 4.4) (optional).

If the structural details and control surfaces (the attributes marked ‘optional’ in Tables 4.4 and 4.5) are specified, it is assumed that the lifting surface has a ‘traditional’ structural layout. This assumption was made to keep the scope of the work manageable. Therefore, if these are specified, the structure is assumed to consist of some or all of the following:

- a ‘lifting surface-structural box’, bound to the front and to the rear by a spar;
- a fixed leading-edge, attached to the front of the box;
- a fixed-trailing edge, attached aft of the box;
- control surfaces, such as slats, flaps, and ailerons, elevators, rudders, and so forth.

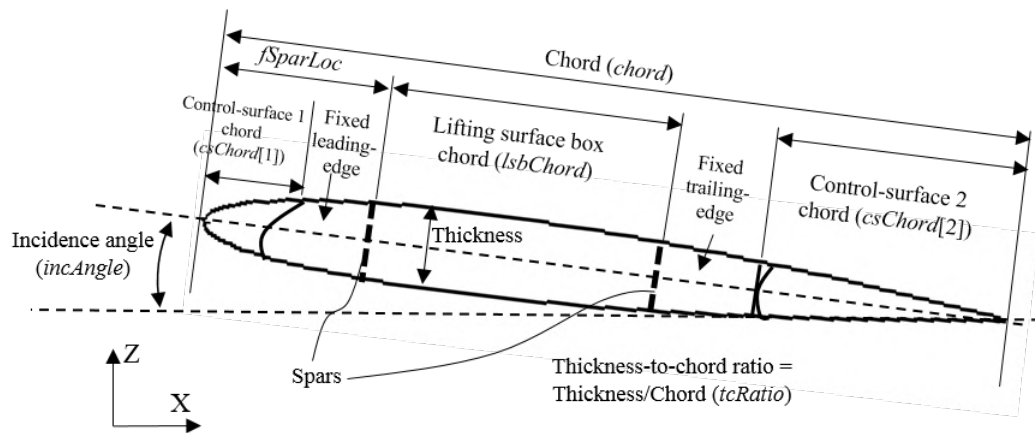


Figure 4.9: Description of the attributes of the lifting-surface station class.
Class and attribute names are shown in *italics*.

4.3.2 Aircraft Design Structure Matrix

The purpose here is to define a design structure matrix that compactly stores important component and topology information for a selected aircraft. Component information could include type, spatial orientation, and symmetry information, amongst others, whereas topology refers to the manner in which the constituent components are connected to each other. This information is to be used to swiftly compare pairs of components of the same category (i.e. fuselages, wings, engines, and so forth) from two selected aircraft, to determine whether they should be inspected in more detail to search for similarity in the subsequent steps.

Note that, to prevent confusion in this discussion, a distinction is made between the ‘type’ and ‘category’ of a component. The ‘category’ of a component refers to what kind of aircraft component it is, such as ‘fuselage’, ‘wing’, engine, etc., whereas its ‘type’ refers to what kind of component it is within the category,

such as a ‘turboprop’ within the ‘engine’ category. Also, note that, although the focus is here on airframe similarity, the DSM would usually include non-airframe components, such as the engines. This is because the way in which these are connected to the airframe components could have a significant impact on airframe similarity across different designs.

For the discussion that follows, let the DSM be called $\mathbf{D}$. Furthermore, let the set of components which constitute the aircraft be denoted as follows:

$$C = \{c_1, c_2, \dots, c_k, \dots, c_n\}. \quad (4.1)$$

The number of components is therefore n . $\mathbf{D}$ is defined here as a square matrix of size $n \times n$, where the entries in row k and those in column k contain information on how the k^{th} component is connected to the other components. The diagonal entries contain information regarding the component in the corresponding row/column itself. This includes orientation, symmetry, and type of the component, amongst others. The off-diagonal entries contain information regarding the location and type of the connection between the component corresponding to the row and the component corresponding to the column of the entry (i.e. topology). A convention was adopted here, in which any spatial attachment information in an entry is always relative to the component of the corresponding row. A nil entry in an off-diagonal cell always indicates that there is no connection between the two corresponding components.

A template for a generic aeroplane DSM was developed, which is shown in Figure 4.10. The format of this DSM is not prescriptive but rather serves as an example of how it could be constructed. The entries shown are field names that contain specific information on the components. The discussion will return to their meaning shortly, but, for now, note that they are specific to particular categories of aircraft components. The category of the component is referred to in this text by an integer ‘category-identifier’, called τ . Table 4.6 lists the major components and the values of τ that were assigned to them. These values correspond to the category vectors shown in Figure 4.10.

Although not shown, extra rows can be added to the DSM for multiple components of the same category (for example, for a twin-fuselage aircraft). Different aircraft will also not necessarily have components from all the categories indicated, so the DSMs for different aircraft could have different sizes.

Component category→										
	1	2	3	4	5	6	7	8	9	10
↓										
1	Sm	Fwa, Za	Ya	Za	Xa, Ya, Za	Ya, Za	FUrt	Ya, Za	Za	Ya, Za
2	Ya	Sm	Ya	N/A	YN	Xa, Y, Za	WUrt	Za	Ya, Za	Xa, Y
3	Za	Za	ST, RX, Sm	Za	N/A	Ya, Za	N/A	Za	N/A	Za
4	Ya	N/A	Ya	ST, RX Sm	N/A	Ya	N/A	Ya	N/A	Ya
5	YN	YN	N/A	N/A	ET, Sm	YN	N/A	N/A	N/A	N/A
6	bl	bl	bl	Za, bl	YN	Sm	NUrt	Ya, Za	N/A	Xa, Ya, Za, bl
7	YN	YN	N/A	N/A	N/A	YN	LGA	N/A	N/A	N/A
8	lw	lw	lw	lw	N/A	lw	N/A	PLO, Sm	N/A	N/A
9	lw	lw	N/A	N/A	N/A	N/A	N/A	N/A	Sm	N/A
10	lw	lw	lw	lw	N/A	lw	N/A	N/A	N/A	Sm

Figure 4.10: Generic aircraft design-structure matrix (DSM) format.

Table 4.6: Aircraft component categories.

Component category identifier (τ)	Description
1	Fuselage
2	Wing
3	Vertical lifting surface
4	Horizontal lifting surface
5	Engine
6	Nacelle
7	Undercarriage
8	Pylon
9	Strut
10	Boom

The meanings of the different field names in the DSM are provided in Table 4.7. In the text that follows, they are printed in `Courier` font, as these are the names used for them in the software implementation. They store information which is

taken from the attributes values of the different objects. Each field is a one-dimensional array of which all the entries are zero, except for a possible single entry of one. The location of the one entry indicates what value is selected for the field (options of which are provided in the third column in Table 4.7). The field can therefore be regarded as a binary number, of which the value refers to the applicable connection or type.

To obtain the complete entry of a cell in the matrix, all the field arrays in the cell are concatenated (in the order that they are presented in Figure 4.10) to form a larger binary number. This entry can be parsed by the techniques presented later to retrieve the stored information again. If the entry is zero, it means that there is no connection between the two corresponding components.

As an example of this technique, it can be seen that in row 1, column 2 (i.e. the fuselage-wing connection), in Figure 4.10, the entry contains FWa (fuselage-wing attachment), followed by Za (the vertical location on the fuselage where the wing is attached). If the fuselage-wing attachment was, say via ‘ring frames’, FWa would equal [0100] and, if the wing was attached to the middle of the fuselage, Za would be [001000]. These are concatenated to form the binary number, $B(2, 3) = [0100001000]$ (with the most significant bit on the right). This would be the entry of the cell and could be converted to a decimal number, if so desired, in which case $D(2, 3) = 66$.

This technique enables a large amount of information to be stored in a single matrix entry.

Table 4.7: Description of Airframe DSM entry fields.

Field	Description	Values
b1	Blended into surface (either yes (1) or no (0))	{0; 1}
ET	Engine type	{Prop, turbo-prop, turbojet, turbofan, CROC, BLI, electric motor-prop; ramjet; scramjet}
FUrt	Fuselage undercarriage retraction types	{fuselage; fuselage podded; wing-fuselage junction} (see Raymer [9, p. 375])
FWa	Fuselage-wing attachment type	{wing-box carry-through; ring frames; bending beam; strut-braced} (see Raymer [9, p. 230])
LGA	Undercarriage arrangement	{Single main; taildragger; quadricycle; bicycle; tricycle or multi-bogey} (see Raymer [9, p. 353])
lw	Dimensionless lengthwise/spanwise location of attachment	{0; 0.25; 0.5; 0.75; 1}
NUrt	Undercarriage retraction into nacelle (either yes (1) or no (0))	{0; 1}
PLO	Pylon orientation	{vertical, horizontal, oblique}
RX	Relative x-location to centre-of-gravity	{front, rear}
Sm	Symmetry	{none; Y; Z}
ST	Stabiliser type	{all movable; trim-movable and control surface; fixed and control surface; fixed}
WUrt	Wing undercarriage retraction type	{wing; wing-podded; wing-fuselage junction} (see Raymer [9, p. 375])
Xa	Streamwise location of attached component	{front; mid; rear; along length}
Ya	Spanwise location of attached component	{far-inner; inner; mid; outer; far-outer; along length}
YN	Either yes (1) or no (0)	{0; 1}
Za	Vertical-location of attached component	{bottom; low; mid; high; top; along length}

Relative spatial information, such as X_a , Y_a , and Z_a , can be obtained automatically by querying the geometry to see where the geometries of the constituent components ‘touch’ (i.e. where they intersect), or could be entered manually by the user.

To summarise the discussion in this section, it can be seen that $\mathbf{D}$, for a given aircraft, is simply an $n \times n$ DSM that contains binary or decimal entries that summarise information on the components and their interrelationship. In this DSM, $\mathbf{D}(i, j)$ refers to the connection between component c_i and component c_j . The DSMs for aircraft a and b , $\mathbf{D}^{(a)}$ and $\mathbf{D}^{(b)}$, are dispatched to the airframe partitioning inference mechanism, which is described next. Note that aircraft a and b , or components from them, are denoted by the superscripts (a) and (b) , throughout the text.

4.4 Airframe Partitioning

In this step, a simple algorithm is provided for parsing the DSMs described in the previous section. The purpose is to determine which pairs of airframe components from the two aircraft under study should be queried for similarity. The output of this step is a ‘comparison matrix’, $\mathbf{CR}^{(\tau)}$, for each category of component, τ , in which the rows correspond to the components of category τ in aircraft a and the columns to those in aircraft b . An entry of one in this matrix indicates that the components in the corresponding row and column will be sent for more detailed similarity assessment, whereas an entry of zero indicates that it will not (and hence is considered to be too dissimilar for any design re-use to be possible).

First, information on components of the same category, $C^{(\tau)} \subseteq C$, for each aircraft is organised together in ‘category-connection’ matrices, denoted by $\mathbf{T}^{(\tau)}$. The size of $\mathbf{T}^{(\tau)}$ is $n \times 3 \times n^{(\tau)}$, where $n^{(\tau)} \leq n$ refers to the number of components of category τ in the specific aircraft. Now, let $m \subseteq k$ be a set of size $n^{(\tau)}$, where k is the ordered set of DSM indices of all the components in the aircraft (i.e. the components in set C), and such that m contains the original ordered DSM indices of the components in $C^{(\tau)}$. Then, for a component $c_p^{(\tau)} \in C^{(\tau)}$ (with $p \in \{1, 2, \dots, n^{(\tau)}\}$ – i.e. p maps to the original indices m), let the columns of $\mathbf{T}^{(\tau)}$ be defined as follows:

$$\begin{aligned} \mathbf{T}^{(\tau)}(\{1, \dots, n\}, 1, p) &= k; \\ \mathbf{T}^{(\tau)}(\{1, \dots, n\}, 2, p) &= \mathbf{D}(\{1, \dots, n\}, m(p)); \text{ and} \\ \mathbf{T}^{(\tau)}(\{1, \dots, n\}, 3, p) &= \mathbf{D}(m(p), \{1, \dots, n\})^T. \end{aligned} \tag{4.2}$$

Therefore, the first column of $\mathbf{T}^{(\tau)}$, for component $c_p^{(\tau)}$, lists all the components in the specific aircraft. The second column contains information on the connections

of these components, relative to component $c_p^{(\tau)}$. Finally, the third column contains information on these same connections, but now relative to the components attached to component $c_p^{(\tau)}$.

$\mathbf{T}^{(\tau)}$ is used in different ways, depending on the component category under consideration. All the fuselages and all wings of the one input aircraft will be compared with those of the other aircraft, regardless of the results in $\mathbf{T}^{(\tau)}$. Rather, in these cases, $\mathbf{T}^{(\tau)}$ is used, along with the object model, during detailed similarity assessment (see Section 4.5) to determine if sub-segments of the fuselages and wings have the same types of components attached to them at the same locations. For all the other categories of components, Algorithm 4.1 is run to determine whether the components of the same category should/should not be compared across the two aircraft.

```

input : Aircraft  $a$  category-connection matrix,  $\mathbf{T}^{(a)(\tau)}$ ; Aircraft  $b$ 
        category-connection matrix,  $\mathbf{T}^{(b)(\tau)}$ .
output:  $\mathbf{CR}^{(\tau)}$  – category  $\tau$  component comparison matrix.
1  $n^{(a),(\tau)} \leftarrow \text{length of } \mathbf{T}^{(a)(\tau)}[1, 1, \dots];$ 
2  $n^{(b),(\tau)} \leftarrow \text{length of } \mathbf{T}^{(b)(\tau)}[1, 1, \dots];$ 
3 for  $p^{(a)} \leftarrow 1$  to  $n^{(a),(\tau)}$  do
4   for  $p^{(b)} \leftarrow 1$  to  $n^{(b),(\tau)}$  do
5     Remove rows in  $\mathbf{T}^{(a)(\tau)}$  where  $\mathbf{T}^{(a)(\tau)}[\dots, 2, p^{(a)}] = 0;$ 
6     Remove rows in  $\mathbf{T}^{(b)(\tau)}$  where  $\mathbf{T}^{(b)(\tau)}[\dots, 2, p^{(b)}] = 0;$ 
7     Sort the rows in  $\mathbf{T}^{(a)(\tau)}$ , such that the entries in  $\mathbf{T}^{(a)(\tau)}[\dots, 1, p^{(a)}]$ 
      are in numerical order;
8     Sort the rows in  $\mathbf{T}^{(b)(\tau)}$ , such that the entries in  $\mathbf{T}^{(b)(\tau)}[\dots, 1, p^{(b)}]$ 
      are in numerical order;
9     Round down the values in  $\mathbf{T}^{(a)(\tau)}[\dots, 1, p^{(a)}];$ 
10    Round down the values in  $\mathbf{T}^{(b)(\tau)}[\dots, 1, p^{(b)}];$ 
11    if  $\mathbf{T}^{(a)(\tau)}[\dots, \dots, p^{(a)}] = \mathbf{T}^{(b)(\tau)}[\dots, \dots, p^{(b)}]$  then
12      // Components to be sent for detailed
      comparison
13       $\mathbf{CR}^{(\tau)}[p^{(a)}, p^{(b)}] \leftarrow 1;$ 
14    else
15      // Components too different to be compared
16       $\mathbf{CR}^{(\tau)}[p^{(a)}, p^{(b)}] \leftarrow 0;$ 
17    end
18  end
19 end

```

Algorithm 4.1: Creating a comparison matrix for components of category τ .

Algorithm 4.1 determines whether component $c_p^{(a)}$ have the same number and type of connections with other components as $c_p^{(b)}$. If this is not the case, the comparison

matrix entry corresponding to $p^{(a)}$ and $p^{(b)}$ ($\mathbf{CR}^{(\tau)}(p^{(a)}, p^{(b)})$) is set to zero, which means that these components are too dissimilar to be considered for commonality. The reason for this is that, if airframe components of the same category across two aircraft are attached 1) to a different number of other components, 2) with different types of attachment mechanisms, or 3) at different locations, the structural considerations and aerodynamic interference characteristics would usually be so different that it would be unreasonable to attempt to make them common. This is different for wings and fuselages, which are usually much larger than the other airframe components. Another exception is made for the undercarriage, where, if the arrangement is ‘tricycle’ for one aircraft and ‘multi-bogey’ for the other, they will still be sent for similarity assessment. This is because gear can be added to a tricycle arrangement to form a multi-bogey arrangement (or vice versa). An actual example of such a case is the A330/A340 family, where the A330 employs a tricycle and the A340 a multi-bogey arrangement. In this case, Airbus removed the centre main undercarriage of the A340, because it was not needed in the A330 [225].

4.5 Airframe Commonality Prediction

In this section, the airframe commonality prediction inference module is presented. An overview of this module is provided in Figure 4.11. As can be seen, the input to this module are the instances of the *Aircraft* class for aircraft a and b , the component comparison matrices that were created as output of the airframe partitioning (discussed in the previous section), and the category- connection matrices for the fuselages, nacelles and wings. These are sent to specific component commonality prediction modules (CPMs), depending on their category.

Five of these CPMs were developed during this research. They are for ‘bodies’ who share similar geometrical properties (such as fuselages and nacelles), the wing, the empennage surfaces (which are horizontal or vertical stabiliser surfaces, and could include canards), the undercarriage, and support components (pylons, struts, and so forth). Some types of airframe components, such as booms and oblique lifting surfaces (which would be found, for example, on ‘V-tail’ aircraft) were not catered for in this research. However, the framework should be easily extendible to include them. Finally, to use these methods, mass distributions must be available for at least the fuselages, wings and empennage lifting surfaces. It is becoming more common to calculate these during conceptual design and software such as NeoCASS [239], or the methods provided by Torenbeek [25] could be used for this purpose.

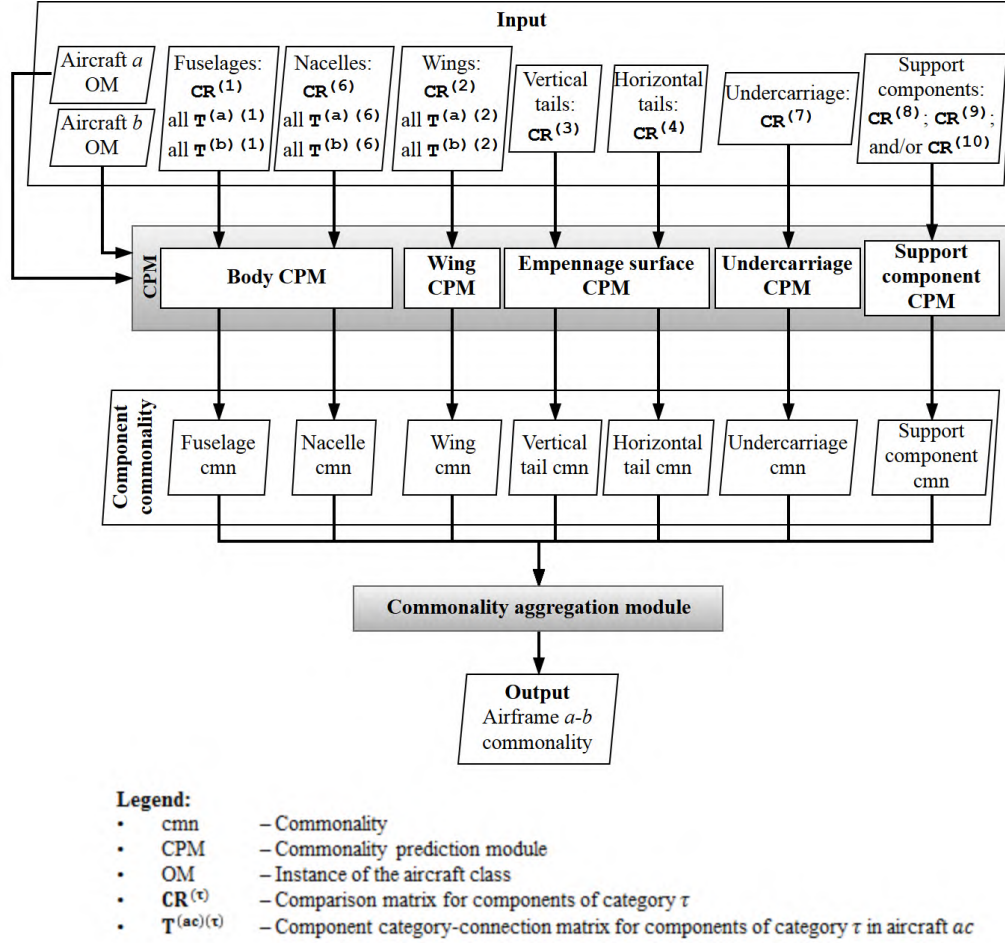


Figure 4.11: Overview of the commonality prediction module.

For each component-pair compared, the modules find the mass of the two components that can be considered common. A predicted ‘mass-commonality’ score is then calculated (for the combination of component i from Aircraft a and component j from Aircraft b) as follows:

$$\Omega_m^{(i,j)} = \frac{m_{cm_{i,j}}}{m_{tot_{i,j}}} = \frac{2m_{cm}}{m_i^{(a)} + m_j^{(b)}}, \quad (4.3)$$

where m_{cm} is the sum of the mass of the sub-segments of a component in one of the aircraft that are found to be similar enough to corresponding sub segments in the other aircraft, such that they can be considered common. Note that it will be likely that the masses of two similar components (candidates for commonality) will usually be slightly different. This difference is assigned to be non-common and not included in m_{cm} . For example, say that we have two components being compared, one of mass $m_i^{(a)} = 5$ kg and another of $m_j^{(b)} = 4$ kg. Next, assume that a sub-segment was found from component i that is similar enough to a sub-segment from j , such that they can be considered common. However, the mass of

the sub-segment from i is 3 kg and the one from j is 2 kg. In this case, the mass of the common segment will be $m_{cm} = 2\text{kg}$ and the commonality score will be:

$$\Omega_m^{(i,j)} = \frac{2m_{cm}}{m_i^{(a)} + m_j^{(b)}} = \frac{2 \times 2}{5 + 4} \approx 0.44. \quad (4.4)$$

The minimum mass of the two common sub-segments is therefore the assigned common mass. The average of the masses of the common sub-segments could also have been taken, but this would not have allowed the difference in mass to be accounted for as ‘additional redesign effort’. In other words, if the average was taken, the two baseline-descendant pairs could appear to have similar merit in terms of commonality. However, because one may have a higher difference in mass for a common sub-segment than the other, taking the average will ‘hide’ that it may actually be less desirable.

A cost commonality score can also be formulated, in a similar fashion as in Equation 4.3, as follows:

$$\Omega_c^{(i,j)} = \frac{c_{cm_{i,j}}}{c_{tot_{i,j}}} = \frac{2c_{cm}}{c_i^{(a)} + c_j^{(b)}}. \quad (4.5)$$

This requires a cost model that can provide estimates of the cost of individual components and their sub-segments. Should such a cost model not be available, the masses of the different components could be weighted by user-defined factors to reflect the relative differences in the ‘effort’ of producing them.

The five different component commonality prediction modules will now be discussed in detail.

4.5.1 Airframe Body Components

The purpose of the body commonality prediction is to identify pairs of longitudinal (x -direction) segments from two parent body objects (from aircraft a and b), which are similar enough to be considered ‘common’. As discussed in Chapter 2.7, this problem can be formulated as a ‘Longest Common Subsequence’ (LCS) problem. The problem is therefore posed formally in this section as such (in an adapted form of the one presented by Bollobás et al. [134]) and algorithms are proposed to solve it.

The inputs to this module are as follows:

- the component comparison matrix, $\mathbf{CR}^{(\tau)}$ (obtained from Algorithm 4.1), where $\tau = 1$ for fuselages and $\tau = 6$ for nacelles;
- the category-connection matrices for both aircraft for the fuselages ($\mathbf{T}^{(a)(1)}$ and $\mathbf{T}^{(b)(1)}$) and nacelles ($\mathbf{T}^{(a)(6)}$ and $\mathbf{T}^{(b)(6)}$); and

- the instances of the classes for the body components to be compared.

For the nacelles, the body commonality prediction algorithms are run only for the entries in $\mathbf{CR}^{(6)}$ that equal one, where the two to be compared will be those in the corresponding row and column of each entry.

4.5.1.1 Discretisation

Let the (geometrical) lengths of the body components from aircraft a and b be referred to as $L^{(a)}$ and $L^{(b)}$, respectively. Now, let the longest of the two bodies be divided by a user-defined natural number, $r \neq 0$, to find a step size, Δx , with which the two bodies can be discretised, i.e.:

$$\Delta x = \frac{\max(L^{(a)}, L^{(b)})}{r} \quad (4.6)$$

and let

$$X = \{0, \Delta x, 2\Delta x, \dots, r\Delta x\} = \{x_0, x_1, \dots, x_r\}. \quad (4.7)$$

X is therefore a set of discretised x -locations. The smaller Δx is, the more accurate the algorithms presented later will be. However, making Δx too small might incur excessive computational cost.

Interpolation can now be performed, using X and the body cross-section attributes values (obtained from the *BodyCrossSection* class – see Section 4.3.1.1), together with their corresponding x -locations, to find the attribute value variation along the x -direction. The attributes of interest for fuselages and nacelles would usually include Hu , Hl , W , *massDistribution*, as well as *effectiveLength* (if the parameterisation technique *bodyXSctShapeParmType*, or the parameters themselves, *bodyXSctShapeParms*, differ between body 1 and 2). For attribute i , at the discretised locations along a specified body, the values obtained through interpolation can then be represented by the following sequence:

$$A_i = \{\alpha_{i,0}, \alpha_{i,1}, \dots, \alpha_{i,r}\}. \quad (4.8)$$

This sequence describes how the value for attribute i changes along the length of the body.

For the shorter body, let the values in the sequence in Equation 4.8 be equal to some ‘dummy’ values for all the values of X that are larger than the length of the shorter body. These dummy values must fall outside the allowable ranges of the respective attribute parameter values.

Before proceeding to the LCS problem statement, it is also required to identify

the x -locations where attachments to specific types of components to the body are present. This information is obtained by parsing the category-component matrices ($\mathbf{T}^{(a)(\tau)}$) and object models for each of the components. The attachments of interest here are those that will imply major ‘cut-outs’, ‘hubs’ (such as for undercarriage), or other features that cannot be simply modified from one design to another. For fuselages, these attached components will usually include the wing(s), engines located within the fuselage, canards, vertical and horizontal tails, and the undercarriage. The x -locations of these can be made identifiable by introducing a parameter, $q_t(x)$, which is zero if there is no attached component, t , at location x , or an integer indicating the type of connection between the body and component t (as assigned in *attachedComponents* – see Section 4.3). Therefore, the following sequence can be defined:

$$Q_t = \{q_{t,0}, q_{t,1}, \dots, q_{t,r}\}, \text{ for } t = \{1, 2, \dots, nq\}, \quad (4.9)$$

where nq is the number of major components attached. nq must be the same for both aircraft, so it will take the value of the body with the most components attached to it. The values in this sequence are set to zero for the shorter body for X values larger than the length of the shorter body.

4.5.1.2 The Body-LCS Problem

An allowable tolerance, $\varepsilon_i \geq 0$, can then be specified (by the user) for the difference in the values of attribute i to be compared. A longest common subsequence (LCS) problem can subsequently be posed as follows:

For $\delta \in \mathbb{N}_0$, find subsequences of X ,

$$X^{(a)'} = \{x_{g_1}, x_{g_2}, \dots, x_{g_l}\} \text{ and } X^{(b)'} = \{x_{h_1}, x_{h_2}, \dots, x_{h_l}\}, \quad (4.10)$$

such that

$$A_i^{(a)'} = \{\alpha_{i,g_1}^{(a)}, \alpha_{i,g_2}^{(a)}, \dots, \alpha_{i,g_l}^{(a)}\}, \quad (4.11)$$

together with

$$Q_t^{(a)'} = \{q_{t,g_1}^{(a)}, q_{t,g_2}^{(a)}, \dots, q_{t,g_l}^{(a)}\}, \quad (4.12)$$

and

$$A_i^{(b)'} = \{\alpha_{i,h_1}^{(b)}, \alpha_{i,h_2}^{(b)}, \dots, \alpha_{i,h_l}^{(b)}\}, \quad (4.13)$$

together with

$$Q_t^{(b)'} = \{q_{t,h_1}^{(b)}, q_{t,h_2}^{(b)}, \dots, q_{t,h_l}^{(b)}\}, \quad (4.14)$$

are the longest common subsequences in $A_i^{(a)}$ and $A_i^{(b)}$, and $Q_t^{(a)}$ and $Q_t^{(b)}$, respectively, for each i and each t , such that:

1. for $1 \leq u \leq l-1$, $g_u < g_{u+1}$ and $h_u < h_{u+1}$;
2. for $1 \leq u \leq l$, $|g_u - h_u| \leq \delta$;
3. for $1 \leq u \leq l$, $\frac{\alpha_{i,h_u}^{(b)}}{1+\varepsilon_i} \leq \alpha_{i,g_u}^{(a)} \leq \alpha_{i,h_u}^{(b)}(1+\varepsilon_i)$;
4. for $1 \leq u \leq l-1$,
 - (a) $(\alpha_{i,g_{u+1}}^{(a)} - \alpha_{i,g_u}^{(a)} > 0 \text{ and } \alpha_{i,h_{u+1}}^{(b)} - \alpha_{i,h_u}^{(b)} > 0)$; or
 - (b) $(\alpha_{i,g_{u+1}}^{(a)} = \alpha_{i,g_u}^{(a)} \text{ and } \alpha_{i,h_{u+1}}^{(b)} = \alpha_{i,h_u}^{(b)})$; or
 - (c) $(\alpha_{i,g_{u+1}}^{(a)} - \alpha_{i,g_u}^{(a)} < 0 \text{ and } \alpha_{i,h_{u+1}}^{(b)} - \alpha_{i,h_u}^{(b)} < 0)$;
5. for $1 \leq u \leq l$, $q_{t,g_u}^{(a)} = q_{t,h_u}^{(b)}$.

The integer, δ , ensures that similarity is only searched for in ‘appropriate’ places (i.e. it prevents similarity to be sought between, for example, a nose of one component and the rear part of another). It also promotes efficiency, as stated by Bollobás et al. [134]. δ can be any user-specified integer, but it must at least be larger than the number of dummy entries in the i^{th} attribute vector, to ensure that the rear end of the shorter body can be compared with that of the longer one. The relational requirements (point 4 in the above problem statement) ensure that ‘diverging’ (i.e. the cross-sectional area of the segment increases as x -increases), ‘converging’ (cross-sectional area is decreasing with x), or ‘constant’ (cross-sectional area is constant with x) segments are compared only with segments with these same characteristics. Note, however, that these relational requirements could be suppressed for selected (non-geometrical) parameters, if so desired.

4.5.1.3 Solving the Body-LCS Problem

Algorithms 4.2 and 4.3 were developed to solve the above problem computationally. Combined, they form an adaptation of a well-known dynamic programming procedure for solving LCS problems, presented in Cormen et al. [135, pp. 351-356]. In Algorithm 4.2, a recursive array is computed, from which the LCS can be constructed. The construction itself is performed in Algorithm 4.3, which provides the x -locations corresponding to the LCS of bodies 1 and 2, namely $X^{(a)'} and $X^{(b)'}$.$

With $X^{(a)'}$ and $X^{(b)'}$, the ‘common segment’ arrays in Equation 4.16 below, of size $d \times 2$, can be constructed. These are lists of the lengthwise locations of the d pairs of segments from body 1 and 2 that have been determined to be ‘similar’ enough to be considered common.

$$\mathbf{CS}^{(a)} = \begin{pmatrix} x_{g_1-1} & x_{g_\mu} \\ x_{g_\nu} & \cdot \\ \vdots & \vdots \\ \cdot & x_{g_l} \end{pmatrix} \quad (4.15)$$

input : $X = \{x_0, x_1, \dots, x_r\}$ (discretised streamwise locations);
 attribute sequences for bodies a and b , namely
 $A_i^{(a)} = \{\alpha_{i,0}^{(a)}, \alpha_{i,1}^{(a)}, \dots, \alpha_{i,r}^{(a)}\}$ and $A_i^{(b)} = \{\alpha_{i,0}^{(b)}, \alpha_{i,1}^{(b)}, \dots, \alpha_{i,r}^{(b)}\}$, as
 well as their corresponding tolerances, ε_i , (all of these for all i);
 the component attachment sequences $Q_t^{(a)} = \{q_{t,0}^{(a)}, q_{t,1}^{(a)}, \dots, q_{t,r}^{(a)}\}$
 and $Q_t^{(b)} = \{q_{t,0}^{(b)}, q_{t,1}^{(b)}, \dots, q_{t,r}^{(b)}\}$ (for all t); and a search length
 limiter, δ .

output: A body LCS recursive array, called $\mathbf{R}$.

```

1  $s \leftarrow \text{length}(X)$ ;
2  $\mathbf{R}[1..s, 1..s] \leftarrow 0_{s,s}$ ;
3 for  $k \leftarrow 2$  to  $s$  do
4   for  $m \leftarrow \max(2, k - \delta)$  to  $\min(s, k + \delta)$  do
5     // First, assign segment to be non-common.
6      $\mathbf{R}[k, m] \leftarrow \max(\mathbf{R}[k - 1, m], \mathbf{R}[k, m - 1])$ ;
7     if for all  $i$ :  $\frac{\alpha_{i,m}^{(b)}}{1 + \varepsilon_i} \leq \alpha_{i,k}^{(a)} \leq \alpha_{i,m}^{(b)}(1 + \varepsilon_i)$  then
8       if for all  $t$ :  $q_{t,k}^{(a)} = q_{t,m}^{(b)}$  then
9         if for all  $i$ :  $(\alpha_{i,k}^{(a)} - \alpha_{i,k-1}^{(a)} > 0$  and  $\alpha_{i,m}^{(b)} - \alpha_{i,m-1}^{(b)} > 0)$  or
           $(\alpha_{i,k}^{(a)} = \alpha_{i,k-1}^{(a)}$  and  $\alpha_{i,m}^{(b)} = \alpha_{i,m-1}^{(b)})$  or  $(\alpha_{i,k}^{(a)} - \alpha_{i,k-1}^{(a)} < 0$ 
          and  $\alpha_{i,m}^{(b)} - \alpha_{i,m-1}^{(b)} < 0)$  then
10          // Segment can be considered common.
            (Overwrite previous assignment of
               $\mathbf{R}[k, m]$ ) .
11           $\mathbf{R}[k, m] \leftarrow \mathbf{R}[k - 1, m - 1] + 1$ ;
12   end
13 end

```

Algorithm 4.2: Finding the body-LCS recursive array.

$$\mathbf{CS}^{(b)} = \begin{pmatrix} x_{h_1-1} & x_{h_\mu} \\ x_{h_\nu} & \cdot \\ \vdots & \vdots \\ \cdot & x_{h_l} \end{pmatrix}; \quad (4.16)$$

In Equation 4.16, $\mu \leq \nu \leq l$, and either $x_{g_\mu} \neq x_{g_\nu}$ or $x_{h_\mu} \neq x_{h_{\nu+1}}$ (or both), except when $\mu = l$. The first column in $\mathbf{CS}^{(a)}$ and $\mathbf{CS}^{(b)}$ therefore displays the x -locations where the common segments commence, whereas the second column shows where they end, for bodies 1 and 2, respectively.

Lists for the non-similar (i.e. non-common) segments ($\mathbf{NCS}^{(a)}$ and $\mathbf{NCS}^{(b)}$) are then constructed in the same way for the x -values in body a and 2 that are not in $X^{(a)'}$ and $X^{(b)'}$, respectively. The common and non-common lists can subsequently be combined and sorted to list all the segments present in bodies a and b . Let these

```

input :  $X = \{x_0, x_1, \dots, x_r\}$  (discretised streamwise locations); and the
         body LCS recursive array,  $\mathbf{R}$ .
output: The  $x$ -locations corresponding to the LCS of bodies 1 and 2,
         namely  $X^{(a)'}$  and  $X^{(b)'}$ .

1  $k \leftarrow \text{length}(X)$ ;
2  $m \leftarrow \text{length}(X)$ ;
3  $u \leftarrow 1$ ;
4 while  $k \neq 1$  or  $m \neq 1$  do
5   if  $\mathbf{R}[k, m] > \mathbf{R}[k-1, m]$  and  $\mathbf{R}[k, m] > \mathbf{R}[k, m-1]$  then
6      $X^{(a)'}[u] \leftarrow x_k$ ;
7      $X^{(b)'}[u] \leftarrow x_m$ ;
8      $k \leftarrow k-1$ ;
9      $m \leftarrow m-1$ ;
10     $u \leftarrow u+1$ ;
11  else if  $\mathbf{R}[k, m] = \mathbf{R}[k-1, m]$  then
12     $k \leftarrow k-1$ ;
13  else
14     $m \leftarrow m-1$ ;
15  end
16 end
17 Reverse order of elements in  $X^{(a)'}$  horizontally;
18 Reverse order of elements in  $X^{(b)'}$  horizontally;

```

Algorithm 4.3: Finding the x -locations corresponding to the LCS of two airframe bodies.

lists be $\mathbf{SS}^{(a)}$, of size $e \times 2$, and $\mathbf{SS}^{(b)}$, of size $f \times 2$. An $e \times f$ sized ‘body commonality matrix’ ($\mathbf{BCM}$) can then be defined, where, if $\mathbf{BCM}(i, j) = 1$, segment i in $\mathbf{SS}^{(a)}$ is common to segment j in $\mathbf{SS}^{(b)}$. Conversely, if $\mathbf{BCM}(i, j) = 0$, i was deemed to be not similar enough to j .

The two segment lists, $\mathbf{SS}^{(a)}$ and $\mathbf{SS}^{(b)}$, along with the body commonality matrix, $\mathbf{BCM}$, are the outputs of the body commonality prediction module. These can be used, together with the mass and cost distributions along the x -direction direction (see Section 4.3.1.1), to compute the mass and cost of the different segments and, hence, the commonality scores with Equations 4.3 and 4.5.

4.5.1.4 Worked Example

A simple example to illustrate the above techniques is provided in this subsection. A MATLAB[®] script for this example is included in Section B.1 in Appendix B. Please refer to Figure 4.12, which shows two aircraft. As can be seen, Aircraft a , has a simple conventional configuration, whereas Aircraft b has a T-tail and have engines attached to the rear fuselage. The fuselages of these aircraft are discretised,

as shown by the grey lines and the corresponding x -values in X .

The attribute values, component attachment sequences, the mass and cost distributions, error tolerances for the fuselage height and width (ε_1 and ε_2), and the search length limiter (δ), are also shown. Note that the entries in the component attachment, mass, and cost sequences, refer to the values for these of the segment directly behind the station indicated by the matching x -value.

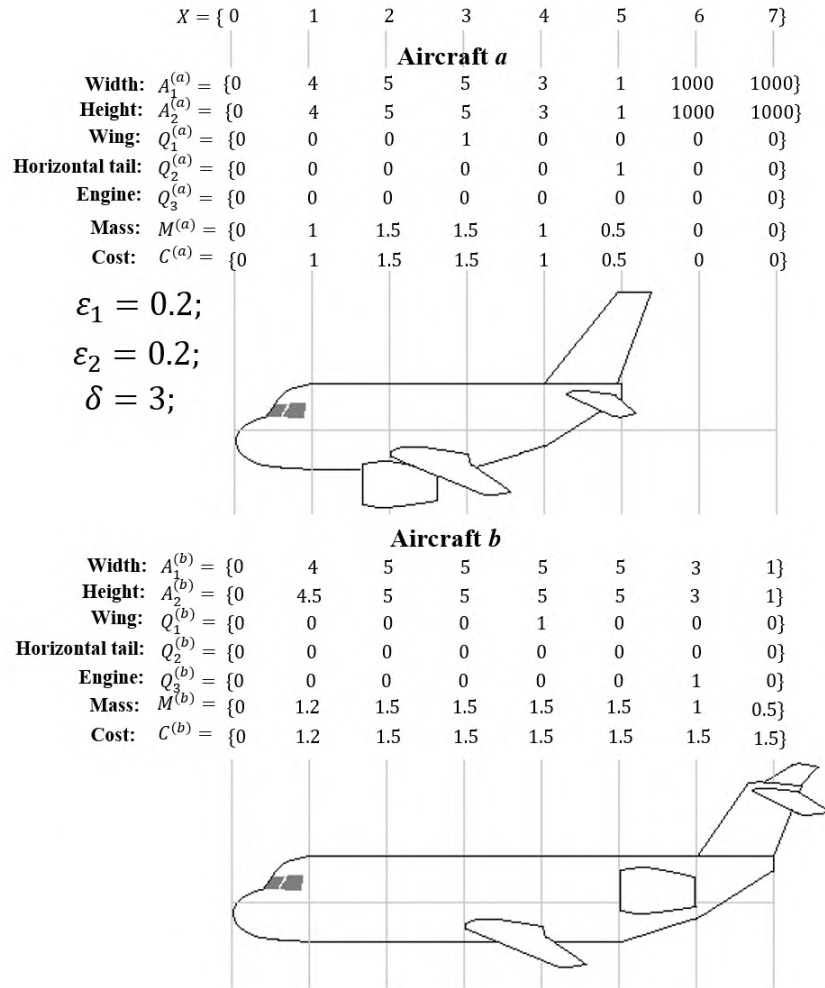


Figure 4.12: Simple example of the input for the prediction of commonality across two fuselages.

The reader can confirm that the recursive array (see Algorithm 4.2) for this example will be as shown in Figure 4.13.

Next, the longest common subsequence, i.e. the x -locations of the common segments can be obtained, as illustrated graphically in Figure 4.14 (also refer to Algorithm 4.3).

		m	2	3	4	5	6	7	8
X			1	2	3	4	5	6	7
k									
			0	0	0	0	0	0	0
2	1		0	1	1	1	1	0	0
3	2		0	1	2	2	2	2	0
4	3		0	1	2	2	3	3	0
5	4		0	1	2	2	3	3	3
6	5		0	0	2	2	3	3	3
7	6		0	0	0	2	3	3	3
8	7		0	0	0	0	3	3	3

Figure 4.13: Recursive array for the example.

		m	2	3	4	5	6	7	8
X			1	2	3	4	5	6	7
k									
			0	0	0	0	0	0	0
2	1		0	1	1	1	0	0	0
3	2		0	1	2	2	2	0	0
4	3		0	1	2	2	3	3	0
5	4		0	1	2	2	3	3	3
6	5		0	0	2	2	3	3	3
7	6		0	0	0	2	3	3	3
8	7		0	0	0	0	3	3	3

Figure 4.14: Finding the longest common subsequence for the example.

As can be seen from Figure 4.14, the x-locations of the similar sections are:

$$X^{(a)'} = \{1, 2, 3\}, \text{ and } X^{(b)'} = \{1, 2, 4\}, \quad (4.17)$$

which means that the common segment arrays are as follows:

$$\mathbf{CS}^{(a)} = \begin{pmatrix} 0 & 1 \\ 1 & 2 \\ 2 & 3 \end{pmatrix} \quad (4.18) \quad \mathbf{CS}^{(b)} = \begin{pmatrix} 0 & 1 \\ 1 & 2 \\ 3 & 4 \end{pmatrix}. \quad (4.19)$$

The non-common segment arrays are therefore:

$$\mathbf{NCS}^{(a)} = \begin{pmatrix} 3 & 4 \\ 4 & 5 \end{pmatrix} \quad (4.20)$$

$$\mathbf{NCS}^{(b)} = \begin{pmatrix} 2 & 3 \\ 4 & 5 \\ 5 & 6 \\ 6 & 7 \end{pmatrix}. \quad (4.21)$$

The segment lists can be constructed from the above as follows:

$$\mathbf{SS}^{(a)} = \begin{pmatrix} 0 & 2 \\ 2 & 3 \\ 3 & 5 \end{pmatrix} \quad (4.22) \quad \mathbf{SS}^{(b)} = \begin{pmatrix} 0 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 7 \end{pmatrix}. \quad (4.23)$$

Using the segment lists, the commonality matrix can be created as follows:

$$\mathbf{BCM} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (4.24)$$

The results from Equations 4.23 and 4.24 are illustrated in Figure 4.15. In this figure, the common segments across the fuselages of the two aircraft are shown in grey, whereas non-common segments are shown in white. The mass and cost for each segment are shown also, along with the final mass and cost commonality scores of $\Omega_M = 0.56$ and $\Omega_C = 0.54$ (as calculated with Equations 4.3 and 4.5). It should be noted that, should an actual aircraft with the configuration of Aircraft 2 be derived from Aircraft 1 (or vice versa), the wing bending moments would likely be substantially different. This will reduce commonality, including in the wing carry-through structure, which will, in turn, affect commonality in the fuselage.

4.5.1.5 Final Notes

The methods presented here are especially useful for bodies that have non-uniform streamwise shapes (i.e. the cross-section shape varies in the x - direction) and aircraft that have dissimilar configurations. This is because the methods will assess where a descendant body will have to change to account for a new shape, where it will need ‘plugs’, and where it will have to be significantly modified to accommodate different components attached to it.

It is important, however, to note that commonality will only be found within the user-specified tolerances. Increasing the tolerances too much might lead to errors in the exact streamwise locations of the boundaries between common and non-common segments developing, as Algorithm 4.2 might assign common segments ‘prematurely’, when more appropriate ones may be available.

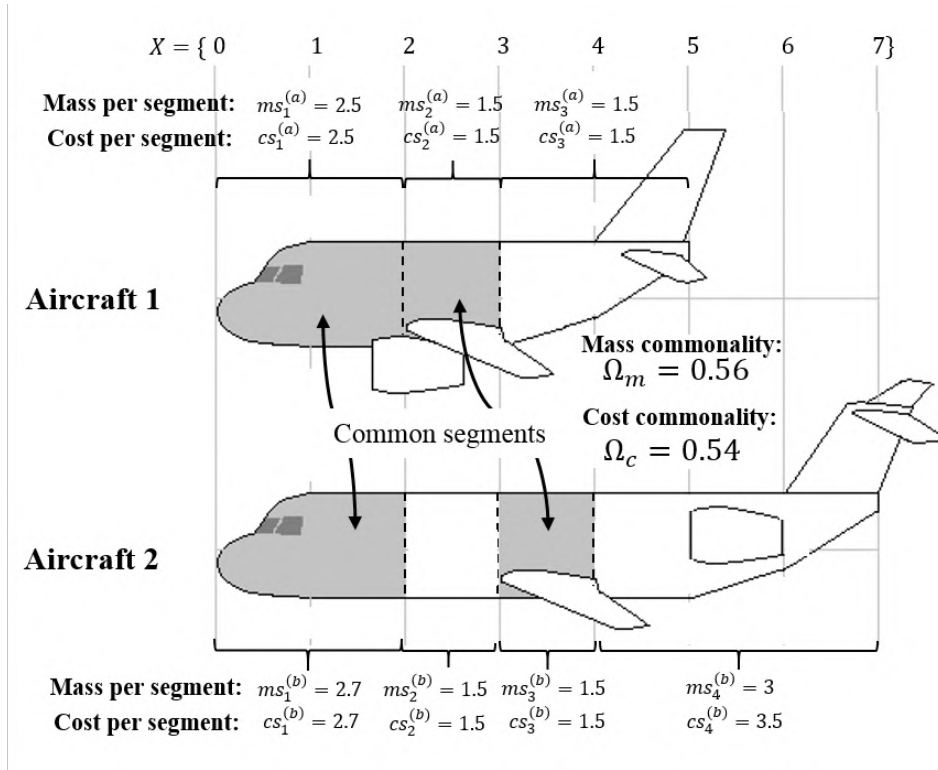


Figure 4.15: Commonality prediction results for the example.

4.5.2 Wings

The same procedure as above could be employed to find similarity between two wings. However, the large changes in chord-length that are sometimes applied to create descendant aircraft implies that any ‘chord-length tolerance’ in a wing LCS problem should be quite relaxed. This may lead to large spanwise errors in the locations of common wing segments (refer the discussion on the tolerance in the previous subsection). Also, as discussed in the case studies in Chapter 3, it is rare that intermediate ‘plugs’ are inserted spanwise in a wing to increase the span. It is far more likely to simply apply a ‘root-insert’, or a ‘tip-extension’.

A different approach is therefore proposed in this section for the wing commonality prediction. Conceptually, it involves finding the ideal length of the wing-tip extension and/or wing root-insert, while keeping other modifications (such as changes in chord length) to a minimum. The wing-box, fixed leading and trailing edges, and control surfaces are then compared across the span of the shorter wing to determine how much commonality there could be in these components across the two wings.

The input to this module is again the component comparison matrix, $\mathbf{CR}^{(\tau)}$ (where, in this case, $\tau = 2$ for wings), the category-connection matrices for all

the wings of both aircraft (i.e. $\mathbf{T}^{(a)(2)}$ and $\mathbf{T}^{(b)(2)}$), and the instances of the classes of the wings to be compared.

To begin the wing commonality prediction, it is noted that changing the sweep or dihedral of a wing appreciably would usually result in the whole wing having to be redesigned. Therefore, before proceeding to a detailed commonality prediction procedure, the quarter-chord sweep-angle ($\Lambda_{\bar{c}/4}$) and dihedral-angle (Γ) of the two wings are compared. If $\Delta\Lambda_{\bar{c}/4} = |\Lambda_{\bar{c}/4}^{(b)} - \Lambda_{\bar{c}/4}^{(a)}| \geq \Delta\Lambda_{\bar{c}/4max}$, or if $\Delta\Gamma = |\Gamma^{(b)} - \Gamma^{(a)}| \geq \Delta\Gamma_{max}$, it is assumed that the two wings are too different to be compared. Here, $\Delta\Lambda_{\bar{c}/4max}$ and $\Delta\Gamma_{max}$ are user-specified maximum allowable differences in quarter-chord sweep and dihedral, respectively. As discussed in Chapter 3, the maximum change in sweep found in the case studies was that across the Airbus A340–300 and A340–600, which was about 1.4° . For all the studies in this research, the allowable change in sweep was therefore set to $\Delta\Lambda_{\bar{c}/4max} = 1.5^\circ$, which is slightly higher. Not much information could be obtained on wing dihedral, but, for argument’s sake, the maximum allowable change in dihedral angle was also set to $\Delta\Gamma_{max} = 1.5^\circ$.

Once the two wings to be compared have passed this ‘sweep/dihedral test’, the detailed comparison can commence, as in the next subsection. However, before proceeding, note that it was decided to not compare aerofoil shape in the commonality prediction procedures. This decision was considered reasonable, because i) trying to predict the exact aerofoil shapes to be used is usually not realistic for the type of studies considered here and ii) the aerofoil shape can be changed (to a certain extent) across generations with comparative minimal effect on the structural layout, as was done with the wings for the newer versions of the Boeing 737 and 747 families (refer to Chapter 3). In such cases, the aerodynamic design is affected, but the structural design less so (should the magnitude and/or distribution of the aerodynamic loads not change too significantly). If desired, the way in which this could be handled in the proposed framework is to break down the development cost of the lifting surface into aerodynamics and structural design cost (amongst others) and reduce only the structural design cost if a given segment is assigned to be common. Similarly, if the same aerodynamic shape is employed, but, say, composites materials are used instead of metals, the cost related to aerodynamic design could be reduced to account for commonality, but not the structural design cost. Doing this, however, requires detailed cost models, which are not available in the public domain. This was therefore not pursued further in the current research.

Similarly, wing thickness was also not considered for commonality prediction (although it is used to predict wing mass for all the studies presented in this thesis). The reason for this decision was that the thickness for a descendant wing will usually be assigned to be very similar to the baseline. Also, because thickness

was indeed used for calculating the wing mass, the effect of dissimilar thickness is indirectly captured in the final mass commonality calculation.

4.5.2.1 Finding Tip-Extensions and Root-Inserts

Figure 4.16 shows the planforms of two wings to be compared (without their carry-through structures). Each of the wings has its own Cartesian coordinate-system, centred at where the leading edge of the wing meets the fuselage. The chord of the baseline wing is represented by $\bar{c}^{(a)}$, whereas that of the descendant wing is given by $\bar{c}^{(b)}$.

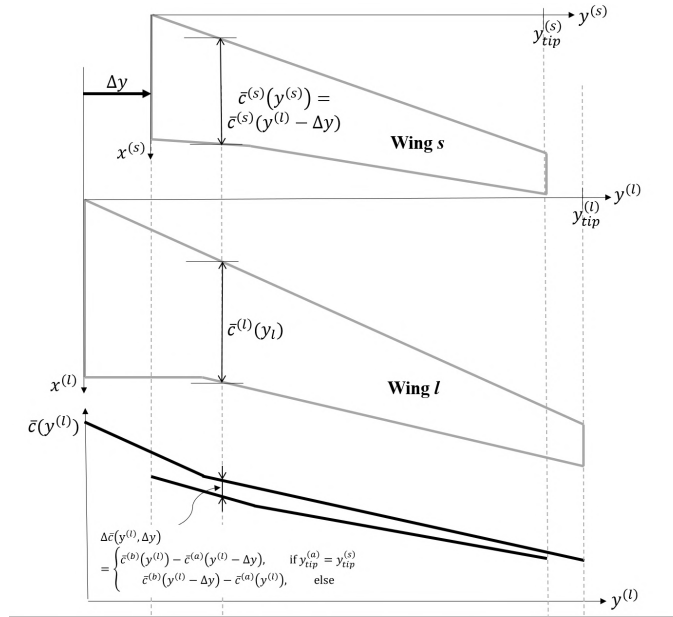


Figure 4.16: Wing planform comparison.

The axis-system of the shorter wing ($x^{(s)}-y^{(s)}$) is allowed to move in a straight line parallel to $y^{(l)}$, between $y^{(l)} = 0$ and $y^{(l)} = y_{tip}^{(l)}$, by adjusting the parameter Δy . Finding the tip-extensions and/or root-inserts, can then be posed as a simple multi-objective optimisation problem, as follows:

$$\begin{aligned}
 &\underset{\Delta y}{\text{minimise}} && f(\Delta y) = w_1 \frac{\Delta y}{y_{tip}^{(l)} - y_{tip}^{(s)}} + w_2 \frac{y(\Delta \bar{c}(y^{(l)}, \Delta y) < 0)}{y_{tip}^{(s)}} + w_3 \frac{dS(\Delta y)}{dS_{max}} \\
 &\text{subject to} && \text{i) } 0 \leq \Delta y \leq y_{tip}^{(l)} - y_{tip}^{(s)}, \quad \Delta y \in \mathbb{R}; \\
 & && \text{ii) } \Delta y \leq y^{(l)} \leq y_{tip}^{(s)} + \Delta y, \quad y^{(l)} \in \mathbb{R}; \\
 & && \text{iii) } \Delta \bar{c}_{max}^- \leq \Delta \bar{c}(y^{(l)}, \Delta y) \leq \Delta \bar{c}_{max}^+, \quad \text{for all valid } y^{(l)}; \\
 & && \text{iv) } w_1 + w_2 + w_3 = 1.
 \end{aligned} \tag{4.25}$$

In the above statement, w_1 , w_2 , and w_3 are user-defined weighting factors for the

three objectives (represented by the three terms in the minimization statement). The first term represents the spanwise length of the root-insert, normalised by the maximum value for Δy . This term was included, because, as discussed in Chapter 3, a root-insert is generally more difficult to implement than a tip-extension. Therefore, during optimisation, it is endeavoured to minimise this.

In the second term,

$$\Delta \bar{c}(y^{(l)}, \Delta y) = \begin{cases} \bar{c}^{(b)}(y^{(l)}) - \bar{c}^{(a)}(y^{(l)} - \Delta y) & \text{if } y_{tip}^{(a)} = y_{tip}^{(s)} \\ \bar{c}^{(b)}(y^{(l)} - \Delta y) - \bar{c}^{(a)}(y^{(l)}, \Delta y) & \text{else} \end{cases} \quad (4.26)$$

is the difference in the chord-length of the descendant wing (b) and that of the baseline wing (a). The numerator of this term, i.e. $y(\Delta \bar{c}(y^{(l)}, \Delta y) < 0)$, represents the sum of the spanwise lengths of segments along $\Delta y \leq y^{(l)} \leq y_{tip}^{(s)} + \Delta y$, in which the descendant chord-length is shorter than that of the baseline. This is normalised by the length of the shorter wing. The second term therefore accounts for the notion that it is more difficult to reduce the chord of a baseline wing than to increase it (again, as discussed in Chapter 3) and that a reduction in chord-length should therefore be minimised.

Finally, in the third term, it is attempted to minimise the difference in the compared planform areas of the two wings, which would vary for different values of Δy . In this term,

$$dS(\Delta y) = \int_{\Delta y}^{\Delta y + y_{tip}^{(s)}} |\Delta \bar{c}(y^{(l)}, \Delta y)| dy^{(l)}, \quad (4.27)$$

and dS_{max} is simply the maximum value of $dS(\Delta y)$ between $y^{(l)} = \Delta y$ and $y^{(l)} = y_{tip}^{(s)} + \Delta y$.

Solving the optimisation problem in equation 4.25 therefore allows one to determine the root-insert length ($b_{root} = \Delta y$) and/or tip-extension length ($b_{tip} = y^{(l)} - (\Delta y + y_{tip}^{(s)})$), which will be required to change the span of a shorter wing to a longer wing. Note, however, that (as discussed in Chapter 3) a reduction in span is difficult. Therefore, if $y_{tip}^{(a)} \geq y_{tip}^{(b)}$, this reduction should be constrained appropriately.

The weightings, w_1 , w_2 , and w_3 will reflect how difficult the designer perceives it to be to increase the length of the root-insert, reduce chord-length at specific locations, or change the chord over the entire compared span.

Constraint (i) simply ensures that the spanwise movement of the shorter wing remains inside the bounds of the span of the longer wing. The second constraint (ii) ensures that the comparison of chord, across the two wings, only takes place

between the spanwise bounds of the shorter wing. Constraint (iii) limits the allowable chord change (relative to the baseline wing) to a user-specified maximum negative difference ($\Delta\bar{c}_{max}^- \leq 0$) and a maximum positive difference ($\Delta\bar{c}_{max}^+ \geq 0$). Appropriate values for $\Delta\bar{c}_{max}^-$ and $\Delta\bar{c}_{max}^+$ could be the maximum changes in chord identified during the case studies, provided in Chapter 3 (see Table 3.23). The fourth constraint (iv), along with the normalizing denominators in the minimization statement, implies that the value of $f(\Delta y)$ will be between 0 and 1.

It will now be shown how commonality can be sought across the shorter wing and the segment of the longer wing that it can be compared with, i.e. within $b_{root} \leq y^{(l)} \leq b_{root} + y_{tip}^{(s)}$, as well as the wing-fuselage structures. The mass distributions for all the structures described next can, for example, be obtained with the methods in Torenbeek [25]. Cost distributions could be based on the mass distribution, or some other in-house methods, if available.

4.5.2.2 Wing-Fuselage Structures

The wing-fuselage structures include the wing-fuselage attachment structure, the carry-through structure (if applicable), and the wing-fuselage fairing. Because the geometries of these components are usually not well-defined during the conceptual design stage, some significant assumptions were made in devising the calculation of the commonality between them. These are:

1. The mass of these components scales linearly with the chord-length of the wing at the location where it meets the fuselage.
2. The mass-distribution is constant stream- and spanwise.
3. If $\Delta y > 0$, there is no commonality in the wing-fuselage structures across the two aircraft. This is because the structural loads and geometry would probably be too different for much re-use of the existing baseline design. Assigning the commonality to be zero in this manner is still quite conservative, however, as the manufacturer will usually have considerable experience with these types of structures. However, to keep the scope of this work manageable, this assumption was adopted.

If $\Delta y = 0$, commonality is computed based on the fractional difference in chord-lengths where the wings meet their respective fuselages, which is given by:

$$\Delta\bar{c}_f = \frac{\Delta\bar{c}(0, 0)}{\bar{c}^{(a)}(0, 0)}. \quad (4.28)$$

The non-common mass is calculated from this, as follows:

$$m_{nc_{wf,i}} = \begin{cases} \Delta\bar{c}_f m_{wf,i}^{(b)} + \left| m_{wf,i}^{(b)} (1 - \Delta\bar{c}_f) - m_{wf,i}^{(a)} \right| & \text{if } \Delta\bar{c}_f \geq 0 \\ (1 + \chi_{wf}) \times \left(-\Delta\bar{c}_f m_{wf,i}^{(a)} + \left| m_{wf,i}^{(a)} (1 + \Delta\bar{c}_f) - m_{wf,i}^{(b)} \right| \right) & \text{if } \Delta\bar{c}_f < 0 \end{cases} \quad (4.29)$$

where the subscript, i , indicates which of the wing-fuselage structures is being considered. The first term in the top condition on the right-hand side of Equation 4.29 attempts to account for the possible larger size of component i of the descendant aircraft. In other words, it is assumed that there will be ‘plugs’ of length $\Delta\bar{c}_f$ in the descendant component, which are not present in the baseline. This first term therefore accounts for the mass of these. The second term (the absolute value) accounts for the possible difference in mass between the remaining, common parts.

For the bottom condition in the equation (i.e. for $\Delta\bar{c}_f < 0$), the terms in the right multiplicand serve the same purpose as the terms in those in for the condition of $\Delta\bar{c}_f \geq 0$. In the left multiplicand, χ_{wf} is a user-defined ‘difficulty factor’, which accounts for the increased effort it might take to reduce the chord. It has the effect of increasing the non-common mass, decreasing the overall commonality. Note that χ_{wf} must be selected such that $m_{nc_{wf,i}} \leq m_{wf,i}^{(a)} + m_{wf,i}^{(b)}$.

The common mass is simply calculated as follows;

$$m_{cm_{wf,i}} = m_{wf,i}^{(a)} + m_{wf,i}^{(b)} - m_{nc_{wf,i}}. \quad (4.30)$$

Also, the common/non-common costs for the wing-fuselage structures ($c_{cm_{wf,i}}$ and $c_{nc_{wf,i}}$) will have exactly the same form as the above equations.

4.5.2.3 Wing-Box Commonality

To calculate commonality for the pair of wing-boxes to be compared (excluding any carry-through structures), they are first discretised spanwise. For the longer wing, only the part of the wing-box between $b_{root} \leq y^{(l)} \leq b_{root} + y_{tip}^{(s)}$ will be discretised. The user specifies an integer, $n > 0$ and obtains the spanwise stations of the wing-box structures to be compared, as follows:

- For the wing-box of the shorter wing (s):

$$Y^{(s)} = \left\{ 0, \frac{y_{tip}^{(s)}}{n}, 2\frac{y_{tip}^{(s)}}{n}, \dots, y_{tip}^{(s)} \right\} = \{y_0^{(s)}, y_1^{(s)}, y_2^{(s)}, \dots, y_n^{(s)}\}. \quad (4.31)$$

- For the longer wing (l):

$$Y^{(l)} = Y^{(s)} + b_{root} = \{y_0^{(l)}, y_1^{(l)}, y_2^{(l)}, \dots, y_n^{(l)}\}. \quad (4.32)$$

The wing-box spanwise mass distributions can then be represented as follows:

- For the baseline wing (a):

$$M_{wb}^{(a)} = \{m_{wb,1}^{(a)}, m_{wb,2}^{(a)}, \dots, m_{wb,n}^{(a)}\}. \quad (4.33)$$

- For the descendant wing (b):

$$M_{wb}^{(b)} = \{m_{wb,1}^{(b)}, m_{wb,2}^{(b)}, \dots, m_{wb,n}^{(b)}\}. \quad (4.34)$$

In the above equations, if the baseline wing is the shorter wing, then $m_{wb,i}^{(a)}$ is the mass of the wing-box segment between $y_{i-1}^{(s)} \leq y^{(s)} \leq y_i^{(s)}$ and $m_{wb,i}^{(b)}$ is the mass of the segment between $y_{i-1}^{(l)} \leq y^{(l)} \leq y_i^{(l)}$, with $\{i = 1, 2, \dots, n\}$. Conversely, if the baseline wing is the longer wing, then $m_{wb,i}^{(a)}$ is the mass between $y_{i-1}^{(l)} \leq y^{(l)} \leq y_i^{(l)}$ and $m_{wb,i}^{(b)}$ is the mass between $y_{i-1}^{(s)} \leq y^{(s)} \leq y_i^{(s)}$.

The cost distributions, $C_{wb}^{(a)}$ and $C_{wb}^{(b)}$, are defined in exactly the same manner.

The commonality can now be calculated in a similar fashion as for the wing-fuselage structures. Note that here it is assumed that the mass and cost per unit (planform) area of the wing-box remains constant in the chord-wise (streamwise) direction. First, the non-common mass for each segment i is calculated as follows:

$$m_{nc_{wb,i}} = \begin{cases} \Delta \bar{c}_{wb,i} m_{wb,i}^{(b)} + \left| m_{wb,i}^{(b)} (1 - \Delta \bar{c}_{wb,i}) - m_{wb,i}^{(a)} \right| & \text{if } \Delta \bar{c}_{wb,i} \geq 0 \\ (1 + \chi_{wb,i}) \times \left(-\Delta \bar{c}_{wb,i} m_{wb,i}^{(a)} + \left| m_{wb,i}^{(a)} (1 + \Delta \bar{c}_{wb,i}) - m_{wb,i}^{(b)} \right| \right) & \text{if } \Delta \bar{c}_{wb,i} < 0 \end{cases} \quad (4.35)$$

Here, $\Delta \bar{c}_{wb,i}$ is the fractional difference in the ‘chords’ (i.e. the distance between the front and rear spars) of the wing-boxes at y_i , obtained as follows:

$$\Delta \bar{c}_{wb,i} = \frac{\bar{c}_{wb,i}^{(b)} - \bar{c}_{wb,i}^{(a)}}{\bar{c}_{wb,i}^{(a)}}. \quad (4.36)$$

$\chi_{wb,i}$, again, is a user-defined ‘difficulty’ parameter that accounts for possible extra effort and expenses involved in reducing the chord length.

The common mass of segment i is then simply:

$$m_{cm_{wb,i}} = m_{wb,i}^{(a)} + m_{wb,i}^{(b)} - m_{nc_{wb,i}}. \quad (4.37)$$

The effects of large and heavy components attached to the wing-box need to be considered as well. For this purpose, the array presented in Equation 4.2 can be employed. This array, along with the discretised locations in Equations 4.31 and 4.32, as well as the instances of the airframe classes of the two aircraft to be compared, can be used to define the following arrays:

$$Q_k^{(ac,\tau)} = \{q_{k,1}^{(ac,\tau)}, q_{k,2}^{(ac,\tau)}, \dots, q_{k,n}^{(ac,\tau)}\}, \quad (4.38)$$

where the value of $q_{k,i}^{(a,\tau)}$ indicates whether a component of category τ is attached at segment i on the wing-box of aircraft ac . The integer k is for when there are more than one component of category τ present. If $q_{k,i}^{(a,\tau)} = 0$, it means there is no category τ component attached at location i . Otherwise, the entry will be that of the corresponding entry in the array in Equation 4.2.

Now, for every segment i , every category τ , and every k , it is checked whether:

$$q_{k,i}^{(a,\tau)} = q_{k,i}^{(b,\tau)}. \quad (4.39)$$

If this condition holds, the common and non-common mass at i , as calculated above, is not affected. If it does not hold, the common-mass is reduced by multiplying it with a user-defined factor, $0 \leq \chi_{wb}^{(\tau)} \leq 1$, which accounts for the difficulty of adding/removing the attachment structure for the specific component attached at location i on the wing-box. The common mass ‘removed’ in this manner is then added to the non-common mass. In other words, if $q_{k,i}^{(1,\tau)} \neq q_{k,i}^{(2,\tau)}$, then

$$m_{cm_{wb,i}}^{(updt)} = \chi_{wb}^{(\tau)} m_{cm_{wb,i}}^{(orig)} \quad (4.40)$$

and

$$m_{nc_{wb,i}}^{(updt)} = m_{nc_{wb,i}}^{(orig)} + m_{cm_{wb,i}}^{(orig)} (1 - \chi_{wb}^{(\tau)}). \quad (4.41)$$

Finally, the common and non-common mass can be calculated as follows:

$$m_{cm_{wb}} = \sum_{i=1}^n m_{cm_{wb,i}} \text{ and } m_{nc_{wb}} = \sum_{i=1}^n m_{nc_{wb,i}}. \quad (4.42)$$

For the longer wing, the mass of the wing box in the root-insert and tip-extension should be added to the non-common mass. The common and non-common cost, i.e. $C_{cm_{wb}}$ and $C_{nc_{wb}}$ can be obtained in the same manner as this mass-commonality prediction procedure proposed above.

4.5.2.4 Other Major Wing Structures

Other major structures of the wing that needs to be compared are the fixed leading- and trailing-edge structures, the high-lift devices (slats and flaps) and ailerons. Here, the same discretisation as in Equations 4.31 and 4.32 is used. The ‘chord’ (i.e. the distance between the front of these components to the rear in the streamwise (x) direction) at i for aircraft a is given by $\bar{c}_{j,i}^{(a)}$, where j indicates which of the major wing-structures it refers to. The common-mass for segment i for component j is then calculated as follows:

$$m_{cm,j,i} = \begin{cases} 2 \min(m_{j,i}^{(a)}, m_{j,i}^{(b)}) & \text{if } \Delta\bar{c}_{j,i} \leq \Delta\bar{c}_j^{(max)} \text{ and } q_{j,i}^{(a)} = q_{j,i}^{(b)} \\ 0 & \text{if } \Delta\bar{c}_{j,i} > \Delta\bar{c}_j^{(max)} \text{ or } q_{j,i}^{(a)} \neq q_{j,i}^{(b)} \end{cases}. \quad (4.43)$$

In this equation, $m_{j,i}^{(a)}$ and $m_{j,i}^{(b)}$ are the masses of segment i for component j for the baseline and descendant, respectively. $\Delta\bar{c}_{j,i}$ is the absolute value of the fractional difference in chords at i , i.e.:

$$\Delta\bar{c}_{j,i} = \frac{|\bar{c}_{j,i}^{(b)} - \bar{c}_{j,i}^{(a)}|}{\bar{c}_{j,i}^{(a)}}. \quad (4.44)$$

Also, $\Delta\bar{c}_j^{(max)}$ is a user-specified maximum allowable difference in the chords. The arrays $q_{j,i}^{(a)}$ and $q_{j,i}^{(b)}$ contain other salient information on component j for the two aircraft. For high-lift devices, the entries in this array would indicate what type of high-lift device it is (for example, ‘triple slotted flap’, ‘Krueger flap’, and so forth). If the entry is zero, it means there is no high-lift device at i . For the fixed leading and trailing edges, the entry would indicate what type of a control surface is attached (if any), as well as if there is undercarriage attached at location i and whether the undercarriage has commonality across the two aircraft.

Therefore, the comparison provided in Equation 4.43 determines, for location i : what the difference in chord-length is for the structure, what specific type of component it is (for high-lift devices) and if undercarriage or a control surface is attached to it (if it is a fixed leading or trailing edge). If the difference in chord-length is larger than a user-specified tolerance, if the type is not the same, if the type of control surface attached is not the same, or if undercarriage is present in one and not the other, the segments at i for the two aircraft are assigned to be completely non-common. This binary assignment of commonality (within the tolerance) is adopted because, unlike the wing-box, the chord of these structures cannot ‘simply’ be extended (except for very small differences), as their shapes are usually much

more complex (to shape the aerofoil). Also, if a different type of flap, for example, is attached, or if the undercarriage is attached at a different location on the descendant aircraft's wing, large modifications are likely for the fixed trailing edge. Setting the commonality to zero is therefore a reasonable (albeit maybe conservative) assignment.

The non-common mass at i is simply:

$$m_{nc_{j,i}} = m_{j,i}^{(a)} + m_{j,i}^{(b)} - m_{cm_{j,i}}, \quad (4.45)$$

and the total common and non-common mass for component j can be calculated as follows:

$$m_{cm_j} = \sum_{i=1}^n m_{cm_{j,i}} \text{ and } m_{nc_j} = \sum_{i=1}^n m_{nc_{j,i}}. \quad (4.46)$$

As was done for the wing-box, the mass of these structures that part of the root-insert and tip-extension (for the longer wing) should be added to the non-common mass. The cost-commonality can, again, be calculated using the same procedures as presented above for the mass-commonality.

4.5.2.5 Overall Wing Commonality

The overall common mass for the wing is simply the sum of the common mass of all the constituent substructures:

$$\begin{aligned} m_{cm_{wing}} = & m_{cm_{ct}} + m_{cm_{wfatt}} + m_{cm_{wffr}} + m_{cm_{wb}} + m_{cm_{fle}} \\ & + m_{cm_{fte}} + m_{cm_{slt}} + m_{cm_{flp}} + m_{cm_{ail}}, \end{aligned} \quad (4.47)$$

where the subscripts are as presented in Table 4.8.

Table 4.8: Subscripts for the constituent wing substructures.

Subscript	Substructure referred to
<i>ct</i>	Carry-through structure
<i>wfatt</i>	Wing-fuselage attachment structure
<i>wffr</i>	Wing-fuselage fairing
<i>wb</i>	Wing-box
<i>fle</i>	Fixed leading-edge
<i>fte</i>	Fixed trailing-edge
<i>slt</i>	Slat
<i>flp</i>	Flap
<i>ail</i>	Aileron

Similarly, the non-common mass is the sum of the non-common mass of all the constituent substructures:

$$\begin{aligned}
m_{nc_{wing}} = m_{nc_{ct}} + m_{nc_{wfatt}} + m_{nc_{wffr}} + m_{nc_{wb}} + m_{nc_{fle}} \\
+ m_{nc_{fte}} + m_{nc_{slt}} + m_{nc_{flp}} + m_{nc_{ail}}
\end{aligned} \tag{4.48}$$

Finally, the total mass is simply:

$$m_{tot_{wing}} = m_{cm_{wing}} + m_{nc_{wing}}. \tag{4.49}$$

One final note about the wing commonality prediction is in order here. Should the geometries of the fixed leading- and trailing-edges, the wing-box, or any of the control surfaces not be available, the same techniques as was applied for the wing-box in Section 4.5.2.3 could be adopted for the whole wing. In such a case, the chords to be compared will be that of the wing itself. The user could then specify a maximum allowable chord change under which a change in chord would be applied to the leading-, or trailing-edge, and another, slightly higher allowable change, under which a chord change would be applied in the streamwise centre of the wing along the span (i.e. the change is applied to the wing-box).

4.5.3 Empennages

The input to this module are the instances of the *LiftingSurface* class (see 4.3.1.2) of the two empennage lifting surfaces selected to be compared by the airframe partitioning module.

The same procedure to predict commonality across two wings, as presented in the previous section, could be applied directly to empennage lifting surfaces. However, as noted in the case studies presented in Chapter 3, when the same empennage layout is employed, the vertical and horizontal tails of the baseline aircraft usually remain unchanged on the descendant, even if there could be a reduction in planform area (a notable exception being the DC-10 to MD-11). If the area has to be increased, it is almost always done by adding a tip extension or root-insert. Another consideration with empennage lifting surfaces is that their exact geometry is usually not that important during conceptual design (Raymer [9, p. 113]) (apart, of course, from the planform area required for stability).

Given these considerations, it was decided to only compare the planform areas of the two empennage lifting surfaces when assessing similarity.

The common mass is then simply estimated as follows:

$$m_{cm_{emp,j}} = \begin{cases} 2 \min(m_{emp,j}^{(a)}, m_{emp,j}^{(b)}) & \text{if } \Delta S_{emp,j}^{(-)} \leq \Delta S_{emp,j} \leq \Delta S_{emp,j}^{(+)} \\ 0 & \text{Otherwise} \end{cases}, \quad (4.50)$$

where j indicates which surface and which type is being compared (i.e. for example, vertical tail, or vertical tail 1, and so forth). $\Delta S_{emp,j} = S_{emp,j}^{(b)} - S_{emp,j}^{(a)}$ is the difference between the areas of the two lifting surfaces. $\Delta S_{emp,j}^{(-)}$ and $\Delta S_{emp,j}^{(+)}$ are user-specified maximum allowable values for the percentage increase and decrease in planform area, respectively. The minimum and maximum values from the case studies provided in Chapter 3 (Table 3.23) could be used for this purpose.

As before, the non-common mass is:

$$m_{nc_{emp,j}} = m_{emp,j}^{(a)} + m_{emp,j}^{(b)} - m_{cm_{emp,j}}. \quad (4.51)$$

4.5.4 Undercarriage

To limit the scope of the research, a detailed comparison method for undercarriages was not pursued. However, a simple method to compare the main undercarriage was proposed, as discussed next (for all studies in this research, it was assumed that the nose gear was common across the two aircraft considered).

As discussed in Chapter 3, changing the length of the undercarriage is exceedingly difficult. Therefore, if the length for the two sets of undercarriages are different a ‘difficulty factor’, with which the commonality could be artificially reduced, could be specified as follows:

$$\chi_{uc} = \left(\frac{\Delta L_{uc}}{\Delta L_{uc}^{(max)}} \right)^{n_{uc}}, \quad (4.52)$$

where ΔL_{uc} is the absolute value of the fractional difference between the lengths of the main undercarriage, $L^{(a)}$ and $L^{(b)}$, of aircraft a and b , respectively, i.e.:

$$\Delta L_{uc} = \frac{|L_{uc}^{(b)} - L_{uc}^{(a)}|}{L_{uc}^{(a)}}. \quad (4.53)$$

The divisor, $\Delta L_{uc}^{(max)}$, and the ‘difficulty power’, n_{uc} , are user-specified parameters. $\Delta L_{uc}^{(max)}$ is a maximum allowable change in length, for which the values found in Chapter 3 could be used. The difficulty power is introduced to reflect the fact that even small changes in length would be difficult to implement. This is done by letting this power be less than zero, which increase the difficulty factor more severely than if it would just have been linear, up to where the change in length reaches the maximum.

The common mass is then calculated as follows:

$$m_{cm_{uc}} = \begin{cases} 2(1 - \chi_{uc}) \min(m_{uc}^{(a)}, m_{uc}^{(b)}) & \text{if } \Delta L_{uc} \leq \Delta L_{uc}^{(max)} \\ 0 & \text{Otherwise} \end{cases}. \quad (4.54)$$

The non-common mass is, again, simply:

$$m_{nc_{uc}} = m_{uc}^{(a)} + m_{uc}^{(b)} - m_{cm_{uc}}. \quad (4.55)$$

As before, the common and non-common costs are calculated in the same manner.

Note, again, that the retraction-type and layout of the two sets of undercarriage compared must be the same (as discussed in Section 4.4). The only exception is when the one aircraft has a tricycle layout and the other a multi-bogey set-up.

4.5.5 Support Components

Support components, such as pylons and struts, are not discretised as was done for the bodies and lifting surfaces. Instead, it is simply checked whether for each attribute:

$$\frac{\alpha_i^{(b)}}{1 + \varepsilon_i} \leq \alpha_i^{(a)} \leq \alpha_i^{(b)}(1 + \varepsilon_i), \quad (4.56)$$

where $\alpha_i^{(a)}$ and $\alpha_i^{(b)}$ are the values for attribute i , associated with the support components being compared across aircraft a and b , respectively. The parameter $\varepsilon_i \geq 0$ is a user-specified tolerance for attribute i .

This limited comparison is not unreasonable, as support components are usually

not as well defined at the early design stages and usually contribute less to the overall mass and cost of the airframe, as compared with the wing and fuselage.

Equations 4.3 and 4.5 are used again to obtain the predicted mass and cost commonality scores.

4.5.6 Commonality Aggregation

To calculate the overall airframe commonality, all the common masses (or costs) for the different components, as calculated with the above methods, are simply summed and then divided by the total mass (or cost) of the two airframes.

It is vital to appreciate, however, that the resulting airframe commonality scores are only meaningful for descendants compared with the *same* baseline (or baselines compared with the *same* descendant). Therefore, the expected overall ‘benefit’ that the commonality provides must be articulated in such a way that the merit of different baseline-descendant pairs can be compared.

There are two ways to do this, which are as follows:

1. The non-common baseline-descendant pair mass (cost), for all the pairs in a study, can be divided by the maximum baseline-descendant pair total mass (cost) found. This provides a number between 0 and 1 that captures the relative difficulty across all the pairs in a study.
2. The cost of common components in the descendant can be reduced, using an appropriate cost model (such as, for example, presented by Markish [104]). The resulting total costs of the descendants can then be used as the comparative measure.

The commonality predictions methods provided in this section work especially well for aircraft with dissimilar configurations. However, as will be discussed later, they can be relatively computationally intensive. A faster, but more limited, technique is presented next.

4.6 Conclusions

Based on the requirement to develop a means for automatic similarity assessment, a computational framework was proposed in this chapter to predict commonality across airframes with dissimilar configurations. The resulting framework could be considered an architecture for a knowledge-based engineering (KBE) application and, as with any KBE application, it employs a knowledge representation and a set of inference mechanisms.

For knowledge representation, a generic aircraft object model and a template for aircraft design structure matrices (DSMs) were introduced. The object model stores salient geometrical, material, and process attributes of the aircraft, whereas the DSM dictates how the different constituent components are attached to each other.

The inference mechanisms proposed include an ‘airframe partitioning module’ and an ‘airframe commonality predictor’. The partitioning module determines which pairs of airframe components from the two selected parent airframes should be compared, based on the type on their type, as well as how they are connected to the other components in the airframe. These pairs are then dispatched to the commonality predictor.

The commonality predictor uses the DSMs, class instances of the airframes, and output from the partitioning module to find common segments across the different components of the two airframes. It was shown how the body (fuselage and nacelle) commonality problem could be stated as a longest common subsequence problem and a an algorithm was propose to solve. It was found that the wing commonality problem could be solved by first posing it as a simple multi-objective optimisation problem, in which the sizes of root-inserts and tip-extensions are found for the longer wing. Thereafter, the substructures of the wings are searched for common segments. Commonality prediction techniques were also proposed for the undercarriage, empennage surfaces, and other support components.

Set-Based Design Techniques for Evolvability Exploration

Chapter Summary: In this chapter, it is demonstrated how the set-based design criteria of interval dominance, maximality, and E-admissibility could be adapted for evolvability exploration studies. This work contributed to achieving the second objective of the research. A mathematical description, followed by a proposed algorithm, is introduced for each criterion. It is demonstrated that E-admissibility is not an appropriate criterion for eliminating near-future designs. A strategy for further elimination of inferior options, by appropriately constraining the evolvability design space, was also proposed. This strategy involves selecting a maximum allowable ‘distance’ from the Pareto frontier for baseline designs and then specifying minimum utility and maximum redesign effort for the possible descendants for each far-future scenario. The union of the sets of far-future designs that adhere to these constraints are retained for further investigation. Together, these techniques constitute a powerful means with which to swiftly but systematically reduce the evolvability design space. This is expected to aid designers to focus their resources on sets of promising designs, while not prematurely converging to a single design.

5.1 Introduction

During conceptual design, a large number of design options are usually generated. In this text, ‘design option’, or simply ‘option’, refers to a single aircraft design and not to a set of designs with similar attribute values, such as, for example, span, or a set of designs with the same configuration. However, sets of these will indeed be considered. As was discussed in the Chapter 2.8, uncertainty is often high during the conceptual design stage, which makes it difficult to determine precisely which of the design options are the most promising. Optimisation often leads to premature selection of design options, which may lead to several time-consuming iterations [15], [16]. These problems are often exacerbated when multiple configurations are considered and, especially, when changeability criteria, such as evolvability, is studied, along with the usual performance criteria.

As explored in Chapter 2.8, set-based design (SBD) is one approach to deal with such problems effectively. This is because set-based filtering criteria enable designers to eliminate provably inferior options analytically, instead of attempting to identify the ‘best’ design too early. However, these are not yet easily applicable to evolvability studies during conceptual design.

In light of this, it is shown in this chapter how SBD decision-making criteria were adapted during this research to enable systematic reduction of the multitudes of design options that could be present in an evolvability exploration study. Each criterion will be stated formally, followed by a procedure and an algorithm (in pseudo-code) to illustrate how it can be implemented. A strategy to further eliminate inferior designs, which involves constraining the evolvability design space in terms of utility, cost, and redesign effort, is also introduced. The work presented in this chapter therefore contributes to achieving the second objective of the research, which was formulated as follows:

Objective 2: To devise new and/or adapt existing set-based design criteria to enable systematic filtering of the evolvability design space.

The layout of the chapter is as follows. First, a description of the input data is provided in Section 5.2. This is followed by the proposed adapted SBD design criteria, namely, ‘interval dominance’ (Section 5.3.1), ‘maximality’ (Section 5.3.2), and E-admissibility (Section 5.3.3). The ‘Constrain and Union’ strategy is then introduced in Section 5.4. Finally, the conclusions follow in Section 5.5.

5.2 Input Description

In this section, the format of the input sets is described. It is assumed (as stated in the scope section of the introduction to this thesis) that there will only be two (consecutive) ‘entry-into-service’ (EIS) timeframes. This was to keep the scope of the research manageable. The first timeframe will be for the ‘near-future’ (EIS1), whereas the second will be for the ‘far-future’ (EIS2). To refine the scope further, only one scenario was considered for the EIS1 timeframe, which is referred to in the text as $S_1 = \{s_{1,1}\}$. The set of possible scenarios for EIS2 is represented as follows:

$$S_2 = \{s_{2,1}, s_{2,2}, \dots, s_{2,n_{s_2}}\} \quad (5.1)$$

where n_{s_2} is the total number of scenarios considered for EIS2. Now, let the set of all scenarios be $S = S_1 \cup S_2$.

Let A be the set of all airframes/design options considered (baselines and descendants). The utility associated with design option $a \in A$ and scenario $s \in S$ can then be calculated based on the stakeholder preferences, social, political, and economic circumstances in s , as well as the attributes of a , by some function $u(a, s)$. Similarly, the cost is calculated by a function, called $c(a)$. Note that it is assumed that cost is not dependent on the scenario. This assumption is applied to simplify the discussion and can easily be relaxed. Also note that if the functions u and c receive a set of aircraft as input, the output is treated as sets of utilities and costs, where the order of the entries depends on the order of the entries in the input aircraft set.

Next, let the design options for EIS1 (the ‘near-future’ or ‘baseline’ options) be grouped into the following set:

$$B = \{b_1, b_2, \dots, b_{n_b}\} | B \subset A, \quad (5.2)$$

where n_b is the number of EIS1 options considered. This set could be divided into smaller sets, in which the constituent airframes share the same configuration, or the same value for some selected attribute (e.g. span). These subsets can be represented as follows:

$$B_i \subset B | i = \{1, 2, \dots, n_B\} \quad (5.3)$$

such that:

$$B_1 \cup B_2 \cup \dots \cup B_{n_B} = B \quad (5.4)$$

and

$$B_g \cap B_h = \emptyset, \forall g, h \in i | g \neq h. \quad (5.5)$$

Similarly, let the design options for EIS2 (the potential ‘far-future’ or ‘descendant’ options) be denoted by:

$$F = \{f_1, f_2, \dots, f_{n_f}\} | F \subset A. \quad (5.6)$$

This set can also have constituent subsets, defined in the same manner as for B in equations 5.3 to 5.5.

Finally, let the ‘redesign effort’ required to change a given baseline, b_i , into a selected descendant, f_j , be calculated by some function $r(b_i, f_j)$ (r would be a set of redesign effort values if one/both of the inputs are sets). The redesign effort could be the actual predicted redesign cost or any other consistent measure of the ‘difficulty’ of changing the baselines into the descendants. Consistency here means that the measure can be used to compare the redesign effort between all the different baseline-descendant pairs in the set of airframes considered. In other words, even if the baselines and/or the descendants of two baseline-descendant pairs are dissimilar, if the redesign measure is higher for the one pair, it must mean that the redesign effort will actually be higher for that pair. This may sound obvious, but there could be measures which are only valid if the same baseline were employed.

Therefore, the commonality prediction method, proposed in the previous chapter, could be used for estimating the redesign effort, but, as discussed in Section 4.5.6, this can be done only if the commonality scores have been ‘translated’ into a normalised commonality score, or if the cost-savings because of commonality have been taken into account to estimate the overall redesign cost for the descendants (to make it consistent).

All the above input information could be entered as arrays. In this case, the arrays for S , A , B , F , and for the subsets B_i , and F_j (see equations 5.3 to 5.5), will retain the same names as the sets. The utility and cost functions, $u(a, s)$ and $c(a)$, will be represented by the arrays U and C , respectively, each of size $|A| \times |S|$. In these arrays, $U[m, n]$ and $C[m]$ represent the utility and cost of airframe option $A[m]$ in scenario $S[n]$. Similarly, the function $r(b_i, f_i)$ will be represented by the array R , of size $|B| \times |F|$. For arrays B_i and F_j , the ‘index’ arrays I_{B_i} and I_{F_j} can be defined. These house the indices of the elements of B_i and F_j that correspond to their original location in arrays B and F (i.e. their supersets). In other words, $B_i[k] = B[I_{B_i}[k]]$ for any $k \in \{1, 2, \dots, |B_i|\}$.

In this chapter, the redesign effort for each baseline-descendant pair will simply be a hypothetical value between zero and one (where the larger the number is, the more rework will be required). Also, it is assumed that the cost, and redesign

effort values for a selected design are deterministic. In other words, it is assumed that, should a selected baseline-descendant pair be developed, the actual values for the cost and redesign effort will be exactly as predicted. Furthermore, it is assumed that the attributes of the design that map to utility are also deterministic. The uncertainty accounted for here is therefore only because of exogenous sources (changing scenarios) and not the inherent endogenous uncertainty associated with the design itself. This assumption will have to be relaxed for these methods to be fully useful in conceptual design, where endogenous uncertainty will have to be accounted for. However, to keep the scope of this research manageable, this assumption was enforced and, in future work, it is aimed to account for the inherent uncertainty in these quantities (see Chapter 7.5).

With the format of the input utility, cost, and redesign effort data now specified, the discussion will next turn to the SBD decision-making criteria.

5.3 Elimination Criteria

In this section, it is shown how the three SBD elimination criteria of interval dominance, maximality, and E-admissibility were adapted for evolvability design space exploration. These could be applied to a set of design options in the order shown in Figure 5.1 to eliminate inferior options systematically. This order is not prescriptive, but, as can be seen, the criteria are increasingly ‘strict’ and use increasingly more computational resources. Stricter criteria will generally find more options that can be eliminated than those which are less strict.

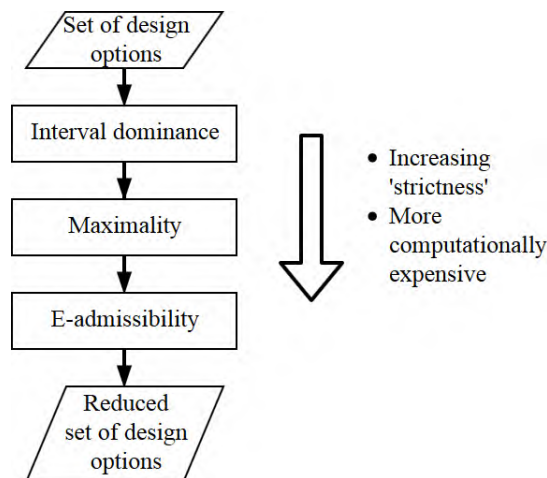


Figure 5.1: SBD elimination criteria.

All these criteria concern determining whether there are designs that are dominated by other designs in terms of utility, cost, and redesign effort (to far-future designs

or from baselines). If this is the case, these designs can be eliminated from the study.

5.3.1 Interval Dominance

The first, and least strict, criterion is ‘interval dominance’ (please refer to Chapter 2.8.1 for a brief introduction to this criterion). The concept of interval dominance was extended in this research to consider not only the utility values ascribable to the attributes of the design options, but also the cost and the interrelationships between the options (i.e. the predicted redesign effort between baselines and descendants). Intuitively, this extended criterion considers whether a baseline design option, b_i , is dominated by at least one other baseline design, b_j , in terms of utility (design option j has higher utility than i), cost (design option j has lower cost than i) in scenario $s_{1,1}$, and redesign effort (i.e. the minimum value of redesign effort between i and all the descendant options is higher than the maximum value of redesign effort between j and all the descendant options). If this were true, baseline i could be removed as an option. Formally, this is stated, for all the baselines considered, as follows:

$$\begin{aligned} & b_i \in B, \exists b_j \in B \\ & [u(b_i, s_{1,1}) < u(b_j, s_{1,1}) \wedge c(b_i) > c(b_j) \wedge \\ & \min(r(b_i, F)) > \max(r(b_j, F))] \Rightarrow \text{remove } b_i \text{ from } B \end{aligned}$$

For far-future options, where multiple scenarios are considered, the following form should be used:

$$\begin{aligned} & f_i \in F, \exists f_j \in F \\ & [\max(u(f_i, S_2)) < \min(u(f_j, S_2)) \wedge c(f_i) > c(f_j) \wedge \\ & \min(r(B, f_i)) > \max(r(B, f_j))] \Rightarrow \text{remove } f_i \text{ from } F \end{aligned}$$

Care must be taken when including cost in the far-future form (for all the criteria presented in this chapter). This is because full development cost could already be captured in the redesign effort. It also depends on whether it is desired to retain the option of creating the far-future designs as clean-sheet designs. If indeed, cost should be included, as the criteria would then evaluate the desirability of a design both as a descendant as well as a clean-sheet design.

A stricter form of this criterion would be to check whether there were at most two equalities instead of the three inequalities in the above statements. In this case, it means that, for example, a design might have the same utility and cost as another but is still dominated by the redesign interval and could therefore be eliminated.

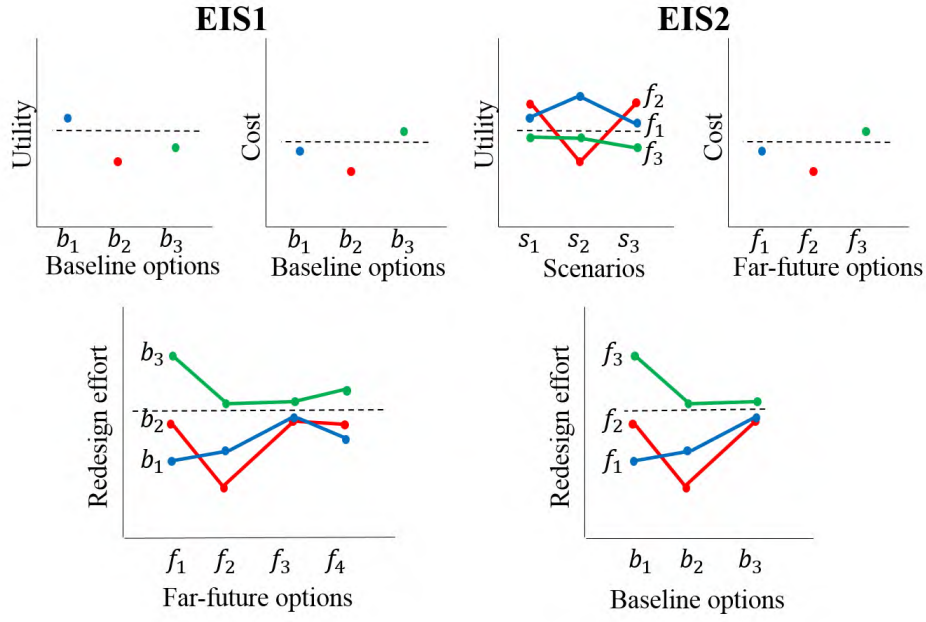


Figure 5.2: Interval dominance applied to baseline design options (left) and far-future options (right).

Both forms are considered for all the criteria in this research.

Figure 5.2 illustrates the interval dominance concept in the preceding mathematical expressions graphically. The three plots on the left of this figure show interval dominance applied to the baselines, where only a single scenario is considered. As can be seen, design option b_3 is dominated by b_1 in utility, cost, and redesign effort to all the descendant options. Option b_3 can therefore be discarded. The plots on the right show the concept applied to the far-future options, where multiple scenarios are considered. Here, the intervals of redesign effort and utility values (associated with the different scenarios) for design option f_1 dominate those of f_3 . Option f_3 can therefore be eliminated.

A simple algorithm is provided below (Algorithm 5.1) to show how this criterion could be implemented computationally. The algorithm shown is for baselines, but is easily extended for far-future options.

In the formulation of this criterion, the words ‘design option’ could be replaced by ‘set of options with the same configuration’, or ‘set of options that share a specific attribute value’. Therefore, if, for example, the set of aircraft with conventional empennages dominated the set of aircraft with t-tails in utility, cost, and all redesign effort values, the set of t-tails could be removed from consideration. Similarly, if a set of aircraft with, for instance, a specific value of span, dominated another set of aircraft with a different value for span, then the latter set could also

```

input : Array of baseline options,  $B$ ; utility array,  $U$ ; cost array,  $C$ ; and
         redesign effort array,  $\mathbf{R}$ .
output: Reduced baseline option array ( $B_r$ )
1  $n_b \leftarrow \text{length}(B)$ ;
2  $B_r \leftarrow B$ ;
3 for  $i \leftarrow 1$  to  $n_b$  do
4   for  $j \leftarrow i + 1$  to  $n_b$  do
5     if  $U[i, 1] < U[j, 1]$  and  $C[i] > C[j]$  and
         $\min(\mathbf{R}[i, ..]) > \max(\mathbf{R}[j, ..])$  then
6       // Option  $b_i$  is dominated by option  $b_j$ 
7       Remove  $B[i]$  from  $B_r$ ;
8     end
9   end
10 end

```

Algorithm 5.1: Interval dominance applied to baseline options.

be removed.

In cases such as these (i.e. with sets rather than individual baseline options), there will be multiple values for utility, cost, and redesign values per far-future aircraft for each set (as there are multiple aircraft in each set). This is illustrated for near-future options in Figure 5.3. As can be seen, for scenario $s_{1,1}$ (EIS1), the minimum utility of all the aircraft in set B_1 (blue) is higher than the maximum utility for all the aircraft in set B_3 (green). Also, the maximum cost in B_1 is lower than is the minimum cost in B_3 . Finally, there is a redesign effort ‘interval’ for each baseline set to each far-future option. This is indicated by the two lines for each set, where the bottom line depicts the minimum redesign value and the top the maximum. It can therefore be concluded for this example that set B_1 dominates set B_3 over the intervals of utility, cost, and redesign effort to all far-future options. Set B_3 can therefore be removed. Formally, this can be formulated as follows (keeping the definition of B_i in equations 5.3 to 5.5 in mind):

$$\begin{aligned}
& B_i \subset \mathbb{P}(B), \exists B_j \subset \mathbb{P}(B), B_i \cap B_j = \emptyset, \\
& [\max(u(B_i, s_{1,1})) < \min(u(B_j, s_{1,1})) \wedge \\
& \min(c(B_i)) > \max(c(B_j)) \wedge \\
& \min(r(B_i, F)) > \max(r(B_j, F))] \Rightarrow \text{remove } B_i \text{ from } B
\end{aligned}$$

An algorithm (Algorithm 5.2) is provided below to apply the criterion for sets. Again, this algorithm is for near-future options, but can easily be extended for far-future designs.

Interval dominance is the least computationally intensive criterion to apply, as it

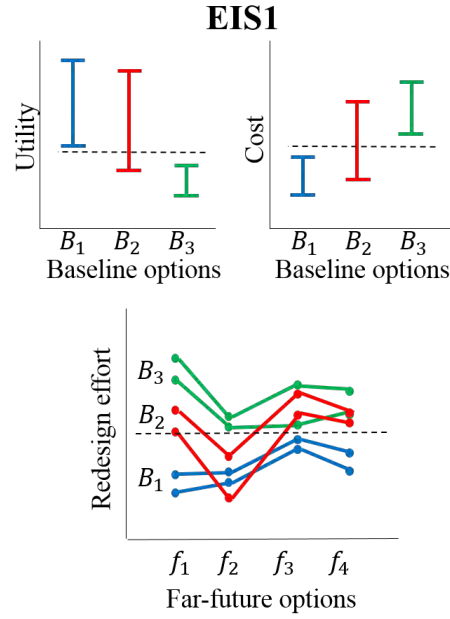


Figure 5.3: Interval dominance applied to configuration or design attribute value sets.

input : Array of the baseline options, B , the n_B sub-arrays of the baseline options, B_i ; the index arrays for B_i , namely I_{B_i} ; utility array, U ; cost array, C ; and redesign effort array, $\mathbf{R}$

output: Reduced baseline option array (B_r)

```

1  $B_r \leftarrow B$ ;
2 for  $i \leftarrow 1$  to  $n_B$  do
3   for  $j \leftarrow i + 1$  to  $n_B$  do
4     if  $\max(U[I_{B_i}, 1]) < \min(U[I_{B_j}, 1])$  and  $\min(C[I_{B_i}]) > \max(C[I_{B_j}])$ 
       and  $\min(\mathbf{R}[I_{B_i}, ..]) > \max(\mathbf{R}[I_{B_j}, ..])$  then
5       // Set  $B_i$  is dominated by set  $B_j$ 
6       Remove  $B_i$  from  $B_r$ ;
7     end
8   end
9 end
    
```

Algorithm 5.2: Interval dominance applied to configuration or design attribute value sets.

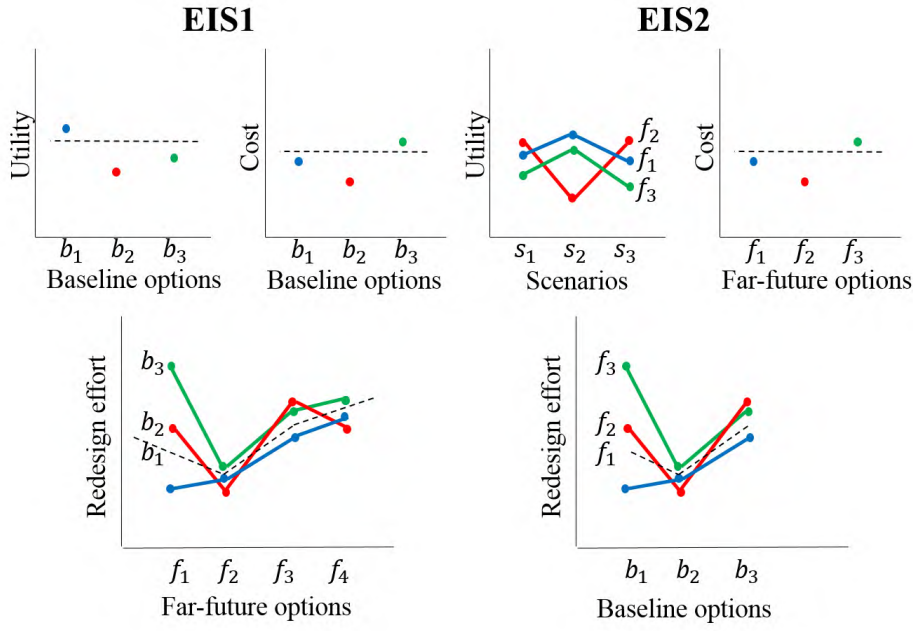


Figure 5.4: Maximality applied to baseline options (left) and far-future options (right).

is only required to find the maximum or minimum values of the utility, cost, and redesign intervals to determine dominance.

5.3.2 Maximality Criterion

In this section, the maximality criterion is adapted for evolvability studies. For a brief introduction to the philosophy behind the criterion, the reader is referred to Chapter 2.8.2.

Figure 5.4, shows a similar situation to what was presented in the interval dominance case in the previous section. As can be seen, for the near-future EIS (the three plots on the left), option b_1 dominates b_3 in utility and cost (in $s_{1,1}$), but the maximum value for the redesign effort to all the far-future options from b_1 is not smaller than the minimum value for b_3 . Therefore, interval dominance will not eliminate any designs in this case. However, the redesign effort with b_1 as baseline is still lower than with b_3 for each separate descendant. Similarly, for the far-future options (the three plots on the right of Figure 5.4), option f_1 has higher utility in each scenario, lower redesign effort from each separate baseline, and lower cost than option f_3 . The maximality criterion is employed to find designs that are dominated in such a way. Mathematically, it can be stated for baseline options as

follows:

$$\begin{aligned}
& b_i \in B, \exists b_j \in B, \forall f_k \in F \\
& [u(b_i, s_{1,1}) < u(b_j, s_{1,1}) \wedge c(b_i) > c(b_j) \wedge \\
& r(b_i, f_k) > r(b_j, f_k)] \Rightarrow \text{remove } b_i \text{ from } B
\end{aligned}$$

For far-future options, the form for maximality is as follows:

$$\begin{aligned}
& f_i \in F, \exists f_j \in F, \forall s_2 \in S_2, \forall b_k \in B \\
& [u(f_i, s_2) < u(f_j, s_2) \wedge c(f_i) > c(f_j) \wedge \\
& r(b_k, f_i) > r(b_k, f_j)] \Rightarrow \text{remove } f_i \text{ from } F
\end{aligned}$$

As with interval dominance, a stricter form of this criterion would be to allow two of the three inequalities in the above statement to be equalities. The algorithm proposed below (Algorithm 5.3) shows how this criterion could be implemented to baseline options. It is easily extended to evaluate far-future options.

<p>input : Array of baseline options, B; utility array, U; cost array, C; and redesign effort array, $\mathbf{R}$.</p> <p>output: Reduced baseline option array (B_r).</p> <pre> 1 $[n_b, n_f] \leftarrow \text{size}(\mathbf{R})$; 2 $B_r \leftarrow B$; 3 $N \leftarrow 0$; // A counter 4 for $i \leftarrow 1$ to n_b do 5 for $j \leftarrow i + 1$ to n_b do 6 if $U[i, 1] < U[j, 1]$ and $C[i, 1] > C[j, 1]$ then 7 for $k \leftarrow 1$ to n_f do 8 if $\mathbf{R}[i, k] > \mathbf{R}[j, k]$ then 9 $N \leftarrow N + 1$; 10 end 11 end 12 if $N = n_f$ then 13 // Option b_i is dominated by option b_j 14 Remove $B[i]$ from B_r; 15 end 16 $N \leftarrow 0$ 17 end 18 end 19 end </pre>
--

Algorithm 5.3: Maximality applied to baseline options.

Finally, maximality is more computationally intensive than interval dominance, as it requires determining whether one design is dominated by another in each scenario and redesign case.

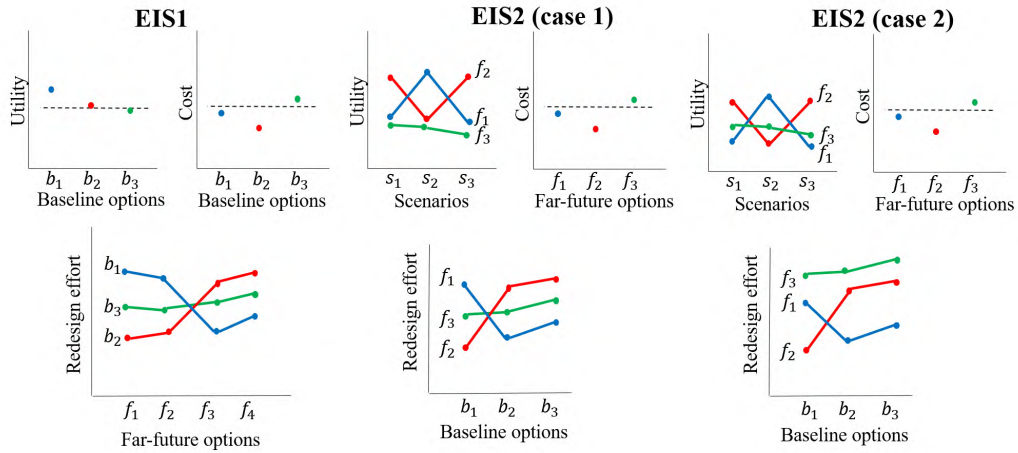


Figure 5.5: E-admissibility applied: near-future (left); far-future (right).

5.3.3 E-Admissibility Criterion

If, after applying interval dominance and maximality, a large number of options remain, E-admissibility can be attempted. Please refer to Chapter 2.8.3 for a brief introduction to this concept. Simply put, E-admissibility determines whether, for a selected design, there is always at least one better option for all possible outcomes. Specifically, it is checked whether, for each possible option, there is at least one other design that has less redesign effort, higher utility, and lower cost in each scenario.

Before proposing a form for E-admissibility for evolvability, it will first be demonstrated that it is not always an appropriate criterion for baseline designs. Consider the three plots on the left of Figure 5.5. These represent the situation for the near-future options. As can be seen, designs b_1 and b_2 dominate b_3 in terms of cost and utility, but not redesign effort to all far-future options. With E-admissibility, b_3 would still be eliminated, as there is always another baseline that has lower redesign cost for every far-future option. However, if b_3 is eliminated and, e.g. b_1 was selected as the design to pursue rather than b_2 , and in the far-future it turns out that f_1 and f_2 are the preferred concepts, the designers would be stuck with b_1 as a baseline, whereas b_3 would have been more appropriate. This happens because the uncertainty regarding which design would be preferred in the far-future has not been resolved at the time a decision regarding which baseline to select has to be made. E-admissibility could still be used to eliminate baseline designs, if it is known which far-future design will be pursued, *before* committing to a baseline. This may be the case for a simple planned derivative.

However, E-admissibility remains useful for eliminating far-future concepts. As

can be seen for the two example far-future cases on the right-hand side of Figure 5.5, for each scenario and for each baseline, there is a far-future descendant that has higher utility, lower cost, and lower associated redesign effort. Here, f_3 can indeed be eliminated, because, no matter which baseline was selected, there will always be a better far-future design to select from. Formally, E-admissibility for evolvability can be stated as follows:

$$f_i \in F, \forall b_k \in B, \forall s_2 \in S_2, \exists f_j \in F [u(f_i, s_2) < u(f_j, s_2) \wedge c(f_i) > c(f_j) \wedge r(b_k, f_i) > r(b_k, f_j)] \Rightarrow \text{remove } f_i \text{ from } F$$

As with the other criteria, this could be made stricter by allowing two of the three inequalities in the above statements to be equalities. Algorithm 5.4 is proposed to apply this criterion within a computational setting.

<p>input : Array of far-future options, F; utility array, U; cost array, C; far-future scenario array, S_2; and redesign effort array, R. output: Reduced far-future option array (F_r)</p> <pre> 1 $[n_b, n_f] \leftarrow \text{size}(R)$; 2 $F_r \leftarrow F$; 3 $n_s \leftarrow \text{length } S_2$; // Number of scenarios 4 for $i \leftarrow 1$ to n_f do 5 $N_1 \leftarrow 0$; // A counter 6 for $k \leftarrow 1$ to n_b do 7 $N_2 \leftarrow 0$; // Another counter 8 for $m \leftarrow 1$ to n_s do 9 Find any option in F, with index g, such that $R[i, k] > R[g, k]$ and $U[i, m] < U[g, m]$ and $C[i] > C[g]$. 10 if <i>Such an option exists</i> then 11 $N_2 \leftarrow N_2 + 1$; 12 end 13 end 14 if $N_2 = n_s$ then 15 $N_1 \leftarrow N_1 + 1$; 16 end 17 end 18 if $N_1 = n_b$ then 19 // A superior option to f_i is always available. 20 Remove $F[i]$ from F_r; 21 end 22 end </pre>

Algorithm 5.4: E-admissibility applied to far-future options.

E-admissibility is at least as computationally resource intensive as maximality, as it is required to determine whether a design option is dominated by any superior

design for each redesign case and in each scenario.

5.4 Constrain and Union Strategy

The concept proposed in this section is not a criterion, as with interval dominance, maximality, and E-admissibility, but rather a procedure that a designer could employ to further filter out design options that are less desirable, based on his/her preferences.

Intuitively, the idea is that the baseline scenario tradespace is constrained by only considering options within a user-specified ‘distance’ from the Pareto front. One technique of doing this is to calculate the ‘Fuzzy-Pareto’ set as described [14]. A minimum utility and maximum allowable redesign effort to the far-future options could then be specified for each future scenario in S_2 , further constraining the evolvability design space. By doing so, there could be descendant options that could meet the utility requirement in at least one scenario, as well as the redesign effort calculated from at least one of the remaining baselines in that scenario, but not necessarily for all the scenarios. However, because it would be a feasible option for at least one of the scenarios in EIS2, it should not be discarded. Therefore, the union is taken of the far-future sets that are created when the constraints are applied.

This technique is depicted in Figures 5.6 to 5.9. Figure 5.6 shows a hypothetical unconstrained evolvability design space. In Figure 5.7, the user has constrained the EIS1 tradespace by, for example, selecting a ‘Fuzzy-Pareto’ set. As can be seen, the designs outside the feasible area are ‘greyed out’. These ‘discarded’ baselines are also shown in grey in the redesign effort vs. far-future option plot. In Figure 5.8, minimum values for utility and maximum values for redesign effort are specified for each scenario, as shown by the green constraint lines. In this case, some of the far-future options (shown in grey) are filtered out, which means that there is no EIS2 scenario where they meet both the utility and redesign constraints.

Figure 5.9 illustrates this more clearly. In this case, only two baseline options depicted by the red and blue points are selected for EIS1. The redesign efforts from these to all the future options can be seen in the lower plot. The constraints that are applied for the far-future scenarios are shown by the green lines. As can be seen, option f_1 meets the utility constraint for scenario $s_{2,2}$, and there is one baseline from which it could be developed for less than the specified redesign effort in this scenario. It is therefore a feasible option for $s_{2,2}$ and should be retained. Option f_2 meets the utility constraint only for $s_{2,3}$, but not the redesign effort. Because there is no scenario in which it meets both the utility and redesign effort

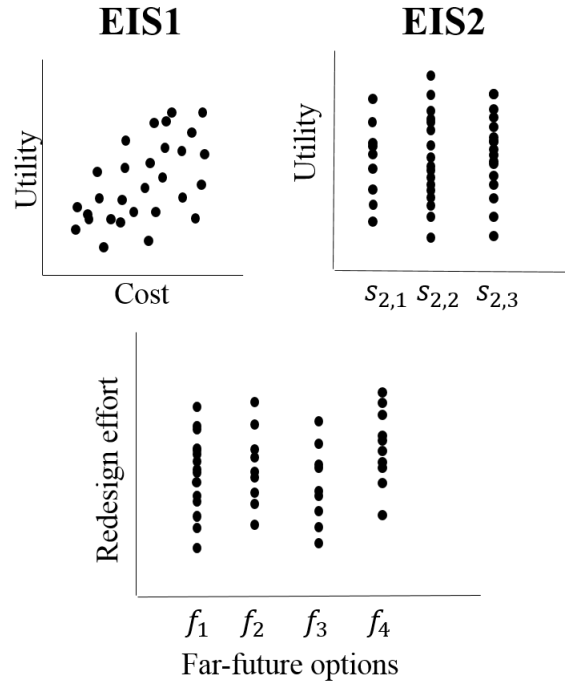


Figure 5.6: Set where no overall dominance is present.

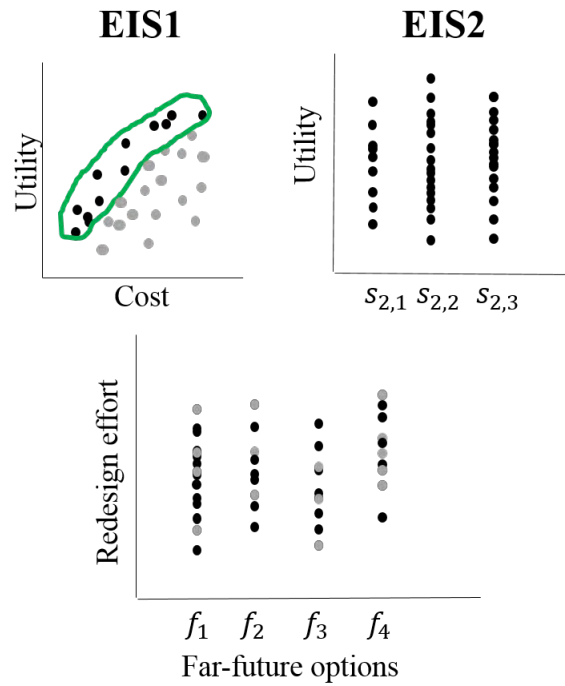


Figure 5.7: Selecting ‘Fuzzy’ Pareto points.

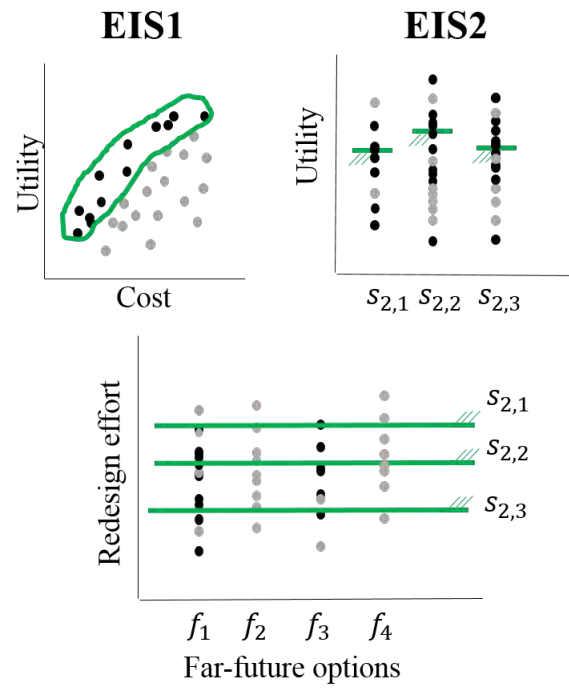


Figure 5.8: Specifying minimum allowable utility and maximum redesign effort.

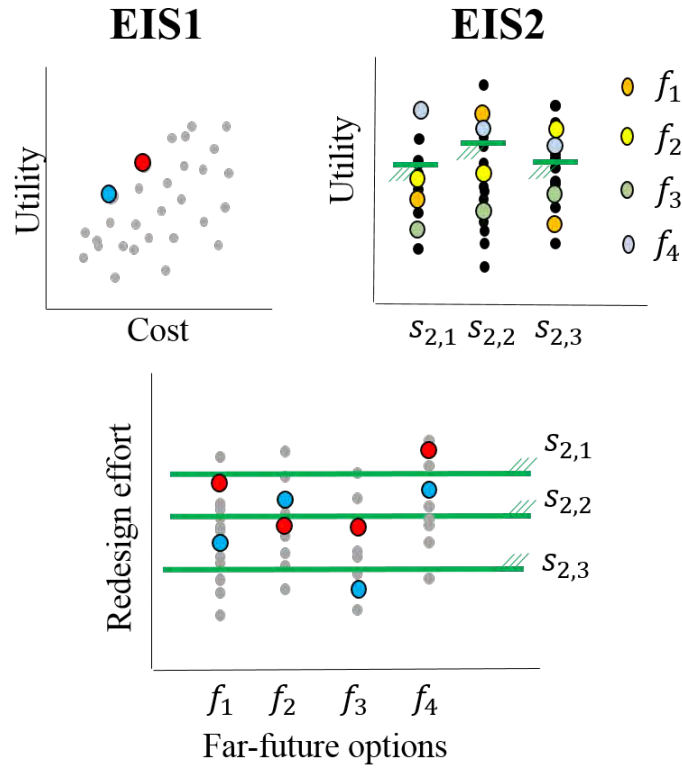


Figure 5.9: Specifying minimum allowable utility and maximum redesign effort with two possible baselines.

requirements, it can be discarded. None of the utility constraints are met with option f_3 ; therefore, it can be discarded too. Finally, f_4 meets the utility criteria for all the scenarios and redesign-effort criteria for one baseline in scenario $s_{2,1}$. It should therefore be retained.

The constrain and union strategy presented here is particularly amenable to interactive exploration. An appropriate software implementation would enable the user to select regions of his/her choosing in the EIS1 scenario, adjust maximum redesign effort and minimum utility values for the different scenarios, and simultaneously obtain visual feedback regarding which far-future designs remain feasible. It could also work the other way around, i.e. the user could select designs of interest in the far-future scenario, and then adjust redesign value, baseline cost and utility to investigate which baseline options remain feasible. Because of time-constraints, such a software implementation was not fully attempted in this research, but some of the intended interactive features are discussed in the next chapter.

5.5 Conclusions

In this chapter, it was shown how the set-based design criteria of interval dominance, maximality, and E-admissibility were adapted for evolvability exploration studies. A mathematical description for each of the criteria was provided, followed by a proposed algorithm. A strategy for further elimination of inferior options, by appropriately constraining the evolvability design space and finding designs that meet these constraints, across all possible far-future scenarios, was also proposed.

It is suggested (although not prescribed) that the criteria be applied in the following order: interval dominance, then maximality, and finally E-admissibility. This is because this order is one of increasing ‘strictness’ (a stricter criterion will likely remove more options than a less strict one would). Another reason is that interval dominance is the least computationally intensive technique, followed by maximality and then E-admissibility. However, as will be discussed in Chapter 6, none of the techniques proved to be particularly computationally expensive during this research. If the designer wishes to reduce the design space further after applying these techniques, the constrain and union strategy could be applied.

It is not advisable to use E-admissibility for eliminating baselines (near-future designs). This is because it is unlikely that a decision would have been made regarding which far-future design to pursue before committing to a near-future design. However, there are exceptions, specifically if the far-future design is planned as a simple derivative, before deciding which baseline to select.

Collectively, these techniques provide the designer with a powerful, systematic

approach to swiftly reduce the number of design options present in an evolvability exploration study by eliminating those that are provably inferior. This allows him/her to focus on promising designs, without converging on a single solution prematurely, which could lead to costly iterations.

In this research, to keep the scope manageable, only two entry-into-service (EIS) timeframes were considered, with only one scenario for the first EIS timeframe. Extending the proposed techniques to account for more timeframes and more scenarios in the initial timeframes, should be relatively straightforward. This is suggested for future research. It was further assumed that the utility, cost, and measure for ‘redesign effort’ for the design options were deterministic. This is unlikely to be the case in actual conceptual design studies, and future work should focus on how to account for uncertainty in these values.

CHAPTER 6

Evaluation

Chapter Summary: In this chapter, the evaluation of the research is presented. The first stage of this evaluation was the support evaluation, in which it had to be determined whether the proposed airframe similarity assessment framework and set-based techniques functioned correctly and whether they produced consistent and accurate results. To this purpose, a demonstration case study involving the design and evolution of a short-range environmentally friendly airliner was developed. The commonality techniques were then applied to these, along with actual historic and current aircraft. The results showed that the techniques function as expected, provide results that were consistent with descriptions of actual commonality achieved in real aircraft derivatives, and could enable a designer to identify possible commonality across aircraft with dissimilar configurations. The results from these studies therefore indicated that Objective 1 was achieved. However, it was found that the execution time of the commonality prediction techniques need to be improved to increase their usefulness. The set-based design criteria were applied to the case study to show how the evolvability space could be progressively pruned of inferior designs. The constrain and union strategy appears especially promising for interactive evolvability exploration. Objective 2 was therefore met. Finally, for the second and third stages of the evaluation (application and success evaluation) a panel of aircraft design specialists from Airbus UK was consulted to obtain feedback regarding the work. The response was positive, indicating that the techniques could potentially add substantial value to their processes.

6.1 Introduction

Several studies were performed to test and evaluate the techniques and algorithms that were proposed during the course of this research. In this chapter, the methods employed, the results, and the conclusions from these studies are provided.

As stated in Chapter 1 (see Section 1.5), the work reported on in this thesis constitutes prescriptive engineering design research and, according to the classification by Blessing and Chakrabarti [18], could be classified as a ‘Type 3’ study. A Type 3 study is where the majority of the work performed is dedicated to producing the ‘design support tool’. In this research, the design support tool consists of the automatic airframe similarity assessment framework and the set-based design techniques for evolvability.

Also recall from Chapter 1 that the ‘Descriptive Study II’ (i.e. the test and evaluation step) of a Type 3 engineering design research undertaking is usually only an initial evaluation, where the main focus is on the ‘support evaluation’ (see Section 1.5). With the support evaluation, it must be demonstrated that the techniques produced function correctly and provide expected and consistent results. The ‘application’ and ‘success’ evaluation studies usually require testing the techniques in an industrial setting, which is beyond the scope of a Type 3 study. However, for this research, limited initial application and success studies were performed by presenting the work to a panel of conceptual aircraft design and modelling and simulation specialists from Airbus UK and obtaining feedback from them.

This chapter is organised into six sections. The first section (this introduction) is followed by Section 6.2, in which the development of a demonstrative case study is described. The purpose of this case study was to develop an example design study to which the techniques proposed could be applied and to illustrate how they could be exploited in evolvability exploration. The results from the case study were employed in all three of the evaluation studies (i.e. support, application, and success). Section 6.3 covers the support evaluation for the airframe similarity assessment framework, followed by Section 6.4 in which the support evaluation for the set-based design techniques are reported on. Chapter 6.5 is devoted to the application and success evaluation studies, which were performed at Airbus UK. Finally, conclusions are drawn and the requirements for future evaluation studies are provided in Section 6.6.

6.2 Demonstration Case Study

6.2.1 Purpose and Overview

To be able to test the techniques and demonstrate how they could be used in early design-stage aircraft evolvability exploration, a demonstration case study had to be developed. To this purpose, a design study for a notional short-range (2,700 nmi), 150-passenger airliner, intended as a replacement for the Airbus A320 and Boeing 737, was created. This category of aircraft was selected, as demand for them is forecasted to be the highest amongst civil transport aircraft for the next 20 years, as per the predictions for single-aisle aircraft in Boeing’s latest global market forecast [240, p. 22]. It therefore serves as a topical exercise.

The goal of the study was to populate both the near-future and far-future entry-into-service (EIS) design spaces with potential designs, upon which the commonality prediction and set-based design techniques could be applied. For the near-future EIS, the clock was wound back to 2015 and, for the far-future, the EIS was set to be within the 2025 to 2035 timeframe. These timeframes were selected, because, as such, the far-future EIS would coincide with the United States National Aeronautics and Space Administration (NASA) ‘N+3’ studies [241], for which much published material exist that could be used for reference purposes.

An N+3 aircraft is one that features an airframe, propulsion, and technologies that are three generations later than the current state of the art (i.e. generation ‘N’) [242]. For the studies consulted, the N+3 aircraft may be feasible by 2035. For this timeframe, NASA has set ambitious goals regarding fuel-efficiency, Nitrogen-Oxide (NOx) emissions, noise, and field length, which are listed for the different generations in Table 6.1.

The cumulative noise is the sum of the ‘lateral’, ‘flyover’, and ‘approach’ noise certification points (see [243] for a thorough explanation of these concepts). The reference point for noise in Table 6.1 refers to the Federal Aviation Administration’s (FAA) Stage 4 regulation [243, 244]. LTO NOx emissions are the landing/take-off nitrogen oxide emissions. The values in Table 6.1 for the LTO NOx emissions are with respect to the International Civil Aviation Organization’s (ICAO) Committee on Aviation Environmental Protection (CAEP) CAEP/6 reference [245]. The percentage reduction in fuel burn indicated in Table 6.1 is with respect to a current state of the art baseline aircraft. For this research, the Airbus A320 was selected as the baseline, because much information is publicly available for this aircraft. The target reductions in field length listed in Table 6.1 proved difficult to achieve with the simulations performed and, as the purpose of this study was only

Table 6.1: NASA N+3 subsonic fixed-wing project metrics (adapted from Ashcraft et al. [242, p. 2])

Aspect	N+1 (2015) ^a Technology benefits relative to a single aisle configuration	N+2 (2020) ^a Technology benefits relative to a large twin aisle configuration	N+3 (2025) ^a Technology benefits
Noise (cumulative below Stage 4)	-32 dB	-42 dB	-71 dB
LTO NO _x emissions	-60%	-75%	Better than -75%
Performance: aircraft fuel burn	-33% ^b	-50% ^b	Better than -70%
Performance: field length	-33%	-50%	Exploit metroplex ^c concepts

^a Technology Readiness Level for key technologies to be between 4 and 6.

^b Additional gains may be possible through operational improvements.

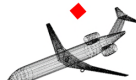
^c Concepts that enable optimal use of runways at multiple airports within the metropolitan areas.

to demonstrate the techniques devised during the research, an arbitrary constraint of 5,200 ft (1,585 m) for field length was assigned. This is still 27% lower than the 2,180 m listed for the A320 in Jenkinson, Simpkin, and Rhodes [26, Data Set A].

6.2.2 Airframe Configurations and Novel Technologies

Several conventional and unconventional configurations and novel propulsion and airframe technologies were considered for this study. The main criterion used when selecting these was that it should be possible to model all the configurations using the same aircraft sizing program (see Section 6.2.4). This was to ensure that the scope of the study remained within the resource limitations of the research.

The configuration-technology combinations (CTCs) are depicted in Figure 6.1. Note that the coloured symbols above the geometry rendering for each aircraft can be used as a guide to identify designs with that specific configuration in the tradespaces that follow. As can be seen in Figure 6.1, for the 2015 EIS two configurations are considered – a conventional ‘tube-and-wing’ (C1) and a T-tail (C3) with rear fuselage-mounted engines (the T-tail can also be considered a conventional tube-and-wing, but the name ‘T-tail’ was used to distinguish it from the more commonly found engine-under-wing configuration). These were modelled with turbofan engine propulsion similar to that found on the Airbus A320neo, Boeing 737 MAX, and Bombardier C-Series, such as the CFM International LEAP engine (the A320neo

	Conventional		T-tail		Noise shielding U-tail		High aspect-ratio strut-supported wing	
2015	C1 		C3 					
	F3STF		F3STF	CROR	F3STF	CROR	F3STF	CROR
2025-2035	No laminar flow	C2a 	C4a 	C4b 	C5a 	C5b 		C6b 
	Passive laminar flow	C2b 	C4c 	C4d 	C5c 	C5d 	C6a 	

F3STF: Future three shaft turbofan

CROR: Counter-rotating open-rotor

The symbols in different colours are a guide to help identify the configurations in the tradespaces that follow.

Figure 6.1: Configuration-technology combinations (CTCs) selected for the demonstration case study.

entered service in 2014, which matches the EIS for this study). The LEAP has a bypass-ratio of between 9:1 to 11:1 [246].

As can also be seen in Figure 6.1, four different airframe configurations were selected for the 2025-2035 EIS timeframe – the conventional tube-and-wing (C2a and C2b); T-tail (C4a-d); a noise-shielding ‘U-tail’ (C5a-d); and the high aspect-ratio (HAR), strut-supported wing (C6a and C6b). To keep the study manageable, only three novel technologies were considered: future three-shaft turbofan (F3STF) engines with ultra-high bypass-ratios ($BPR = 18$), modern ‘propfan’ (also called ‘counter-rotating open-rotor’ or ‘CROR’) engines, and passive laminar flow for the wings. In addition, all the configurations for both the 2015 and 2025-2030 EIS timeframes were to make use of the maximum amount of composite materials in their airframes. These CTCs were selected, because it was expected that they would lead to noticeable trade-offs in fuel burn, cruise speed, noise, and cost results.

The modelling and simulation of these combinations are presented in detail in Section 6.2.4, but, first, the selection of the timeframe scenarios and the development of their associated utility functions are explained.

6.2.3 Scenario Descriptions and Utility Functions

A detailed scenario planning and technology road-mapping exercise was documented by Northrop Grumman for their N+3 proposal, namely the “Subsonic Fixed Wing Silent Efficient Low-Emissions Commercial Transport” (SELECT) (see Bruner et al. [247]). In that study, three main possible scenarios for the 2035

timeframe were identified. These were adopted for the current research and are summarised below [247, Ch. 4]:

- Scenario 1: Bright Bold Tomorrow (BBT). The economy is flourishing and there is widespread ‘point-to-point’ transportation available, meaning larger aircraft must be able to operate from smaller airports and shorter runways. There is also, however, significant emphasis on reducing the impact of aviation on the environment. The author added to this description an increased demand for passenger comfort and convenience (i.e. the need for fast travel and spacious cabins) and a higher willingness from airlines to pay more for aircraft incorporating superior technologies (that reduce fuel-burn, noise, and so forth).
- Scenario 2: Not in My Backyard (NiMBY). Air transportation is growing slowly but remains important. The fast-growing populations around airports are becoming increasingly intolerant to noise and NOx emissions and the focus in this scenario is therefore on reducing these.
- Scenario 3: King Carbon (KC). Deleterious effects of climate change are starting to take its toll and carbon emissions are taxed heavily around the world. Air transportation is especially severely affected by the ensuing rise in fuel prices and the focus of the airframe manufacturers is therefore mainly on improving fuel efficiency.

For the current study, these scenarios were called ‘BBT 2035’, ‘NiMBY 2035’, and ‘KC 2035’. Only one scenario was selected for 2015, and this was devised as similar to BBT, but with less strict field-length requirements. It was called ‘BBT 2015’.

In order to determine the relative value attributed to the different designs by stakeholders in each scenario, a utility function was devised as a simple weighted sum, as follows:

$$u = w_F S_F + w_N S_N + w_C S_C \quad (6.1)$$

This function was adapted from the ‘System-Effectiveness Rating’ metric calculation in the SELECT report by Northrop Grumman [247, p. 14]. Here, u is the utility for a selected design, which should be between 0 (not acceptable) and 1 (the best available). If a utility score was calculated to be equal to or less than zero for a scenario, the associated aircraft was not considered to be an option in that scenario. The coefficients w_F , w_N , and w_C are weighting factors for fuel burn, noise, and passenger comfort, such that

$$w_F + w_N + w_C = 1 \quad (6.2)$$

Table 6.2: Scenario utility calculation weighting factors and constraints.

	BBT 2015	BBT 2035	KC 2035	NIMBY 2035
w_F	0.50	0.33	1	0.45
w_N	0.50	0.33	0	0.45
w_C	0	0.33	0	0.10
Field-length constraint [m]	2080	1580	1830	1830

The values for these are different in each scenario, where they reflect the relative importance of the aspects they weigh (fuel burn, noise, and) in the selected scenario. The values selected for the current research are hypothetical and for demonstration purposes only. They are listed in Table 6.2, for each scenario, along with the field-length constraint and reference noise limits.

S_F , S_N , and S_F are ‘System Effectiveness Ratings’ (SERs) for fuel burn, noise, and passenger comfort, respectively. As with utility, the values for these should be between 0 (not acceptable) and 1 (the best available) for the associated design to be considered in a selected scenario. For fuel burn, the SER for a selected design option is calculated as follows:

$$S_F = \begin{cases} \frac{\Delta BF - \Delta BF_{min}}{\Delta BF_{max} - \Delta BF_{min}} & \text{if } \Delta BF > 0 \\ 0 & \text{else} \end{cases} \quad (6.3)$$

where $\Delta BF = BF_{A320} - BF$, BF is the predicted block-fuel burned by the selected design, and BF_{A320} is the block-fuel burned by the A320 on a nominal design mission (range of 2,700 nmi, with a payload of 14,250 kg [26, Data Set A]). The reference block-fuel was calculated to be 33,882.4 lb (15,370 kg), as is discussed in the next section. ΔBF_{min} is the minimum positive block-fuel reduction amongst all the designs considered in the study, whereas ΔBF_{max} is the maximum attained.

The SER for noise was calculated with the following equation:

$$S_N = \begin{cases} \frac{\Delta N - \Delta N_{min}}{\Delta N_{max} - \Delta N_{min}} & \text{if } \Delta N > 0 \\ 0 & \text{else} \end{cases} \quad (6.4)$$

where $\Delta N = EPNL_{Limit} - EPNL$, and $EPNL$ is the cumulative ‘effective perceived noise level’ for each aircraft. $EPNL_{Limit}$ is the calculated cumulative noise limit, based on FAA Chapter 4, as calculated by the methods in [243]. ΔN_{min} is the minimum positive noise reduction found amongst all the designs considered, whereas ΔN_{max} is the maximum.

Finally, the SER for passenger comfort was calculated with the following expression:

$$S_C = \frac{C_1 + C_2}{2} \quad (6.5)$$

where

$$C_1 = \begin{cases} 1 & \text{if } \Delta M_{Cruise} > 0.75 \\ 0.001 & \text{if } \Delta M_{Cruise} < 0.75 \text{ and current scenario is not BBT} \\ 0 & \text{else} \end{cases} \quad (6.6)$$

and

$$C_2 = \begin{cases} 1 & \text{if } W_{Fus} > 4 \text{ m} \\ 0 & \text{else} \end{cases} \quad (6.7)$$

M_{Cruise} is the cruise Mach number and W_{Fus} is the width of the fuselage. This indicates that much emphasis is placed on convenience (speed) and physical comfort (large cabin space).

Note that, if any of these SERs for a design were calculated to be equal to zero for a particular scenario, the design was not considered as an option in that scenario. In addition, if a design did not meet the take-off and landing field length constraint for a scenario, it was also removed from consideration in that scenario, regardless of the values of the SERs.

6.2.4 Modelling and Simulation

The modelling and simulation of the concepts and scenarios are briefly described in this section. First, an overview of the computational workflow is provided, along with descriptions of the main (high-level) computational models. After this, a short discussion is provided on how the overall workflow (called ‘aircraft sizing, utility, and cost calculator’) was calibrated using published data for the Airbus A320. More detailed descriptions of the models employed are provided in Appendix C, together with important assumptions made and references used.

6.2.4.1 Computational Workflow Overview

Many different computational models were employed in this study. The main aircraft sizing programme employed was the NASA Flight Optimization System (FLOPS) [11], Release 8.23. FLOPS is a comprehensive conceptual aircraft sizing and performance programme that enables many different types of aeroplanes, airframe configurations, propulsion options and technologies to be modelled and

simulated. It contains modules that estimate weights, aerodynamics, mission performance, and perform take-off and landing analyses, amongst others. Unfortunately, FLOPS was not fully adequate for all the calculations needed for this study and several additional models were created, which are discussed further on.

All the models employed were implemented in AirCADia Explorer, a model-based conceptual design tool developed by the Advanced Engineering Design Group at Cranfield University [35, 248]. This software contains numerous in-built enablers for optimisation, interactive design space exploration, uncertainty quantification, and set-based design, amongst others. For the current work, AirCADia was specifically employed to assemble and execute the computational workflow for the different designs considered. It was originally also endeavoured to exploit the advanced visualisation capabilities of AirCADia to perform the evolvability exploration, but this would have required significant modification of some of the in-built algorithms, which fell outside the scope of this work.

A diagram representing an overview of the computational workflow to evaluate the designs is shown in Figure 6.2. This workflow, which calculates mass properties, cost, field-length performance, and scenario utility values for a selected design, was run once for each set of inputs representing an individual aircraft design (the ranges for the input variables are provided slightly further on). Only a brief description of each of the models and input/output/intermediate variable shown in Figure 6.2 are provided here. Explanations of the main models shown in Figure 6.2 are provided in Table 6.3, whereas the overall input, output, and intermediate variables are summarised in Tables 6.4, 6.5, and 6.6, respectively. The reader is referred to Appendix C for more detailed descriptions, along with relevant assumptions made and references employed.

It is important to emphasize here that the models employed are for demonstration purposes only. Several limitations and assumptions made should be kept in mind when interpreting the results. Among the most important of these are as follows:

- The cost model employed was developed by Markish [104], based on development data for the Boeing 777 and other publicly available cost estimation methods. It is therefore likely not generally applicable and possibly does not even provide a correct estimate of the actual 777 development cost (as admitted by Markish [p. 58]). However, it was useful for the purposes of this study, because it provides normalised cost values (cost per unit mass) for specific airframe components, systems, and propulsion. Furthermore, it also provides rough estimates for a percentage cost reduction for when these components, systems, or propulsion, or sub-parts of them, are common with an already-developed baseline (which was the primary reason this model was

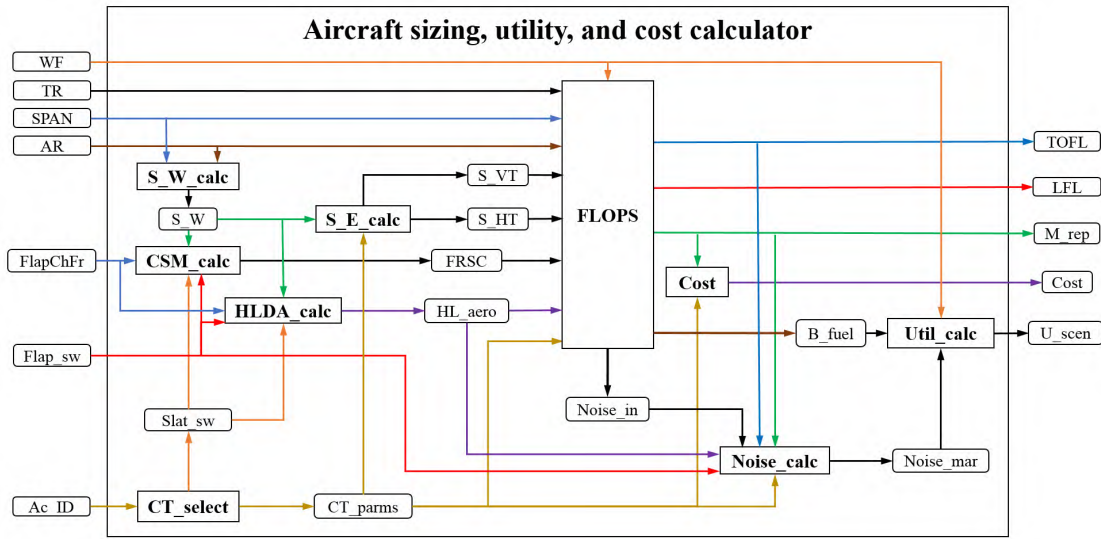


Figure 6.2: Demonstration case study computational workflow. Computational models are represented by the right-angled blocks with model names in boldface, whereas variables are represented by the rounded blocks.

Table 6.3: Main computational models in the demonstration case study workflow.

Model	Description	Main references
FLOPS	Aircraft sizing and performance calculation.	McCullers [11]
CSM_calc	Control surface mass calculator. Estimates the overall control surface mass for different flap/slat combinations (single-, double-slotted, etc.).	Rudolph [221, pp. 118-121]
HLDA_calc	High-lift device aerodynamics calculator. Predicts the maximum lift coefficient and associated drag coefficient for different flap/slat combinations.	Raymer [9, pp. 427-429]
CT_select	Configuration-technology combination selector. Produces the specific input required for sizing, cost and noise calculations for a selected input CTC (C1, C2, etc. See Figure 6.1).	N/A. Refer to the text for more details on this model.
Cost	Calculates non-recurring development and recurring (manufacturing) cost. Note that systems and propulsion costs are excluded.	Markish [104, pp. 51-68]
Noise_calc	Estimates acoustic noise emissions and provide these as noise margins with respect to the FAA Chapter 4 limits.	Michel [249]
Util_calc	Utility calculator. Calculates the utility values for each design (see Eq. 6.1) for all scenarios.	N/A. See Section 6.2.3.

Table 6.4: Overall computational workflow input.

Name	Description
WF	Maximum fuselage width.
SPAN	Wing span.
AR	Wing aspect-ratio.
TR	Wing taper-ratio.
FlapChFr	Ratio of flap chord to wing chord.
flap_sw	Flap switch (1 = single-slotted fowler, 2 = double-slotted fowler, 3 = triple-slotted fowler).
AC_ID	Configuration-technology combination ID (C1, C2, etc. See Figure 6.1).

Table 6.5: Overall computational workflow output.

Name	Description
TOFL	Take-off field length.
LFL	Landing field length.
M_rep	Mass report. Includes Maximum take-off weight (MTOW), as well as airframe component and system masses.
Cost	Cost breakdown array.
U_scen	Utilities of the selected design for each scenario, i.e. $u_{BBT2015}$, $u_{BBT2035}$, $u_{NiMBY2035}$, and u_{KC2035} .

Table 6.6: Computational workflow intermediate variables.

Name	Description
slat_sw	Slat switch (0 = not present, 1 = present). The slat is not present for natural laminar flow configurations in this case.
S_VT	Vertical tail planform area.
S_HT	Horizontal tail planform area.
S_W	Wing planform area.
FRSC	Surface controls mass.
HL_aero	High-lift aerodynamics. Lift and drag polars for high-lift configuration for take-off and landing.
CT_parms	Configuration-technology combination-specific parameters. These are parameters specifying configuration- and technology-dependent information, such as wing, empennage surface, and engine locations; propulsion performance and characteristics; wing sweep, percentage laminar flow area, and so forth.
Noise_in	These are airframe, propulsion, and performance parameters needed to calculate acoustic noise emissions.
B_fuel	Mission block fuel, as calculated by FLOPS.
Noise_mar	FAA Chapter 4 noise margins.

chosen instead of the in-built FLOPS cost methods). The model could therefore be used to compare designs in an ordinal manner, but the actual cost values should be treated as meaningless. It should be noted that true development/manufacturing costs for airliners are usually not publicly available and cost-estimation is an extremely complex activity, which fell outside the scope of this research. In light of this, the use of this model as-is was deemed reasonable for the purposes of the work presented here. To account for the higher cost expected for novel technologies, the cost per mass values in the model was simply doubled for airframe components that are unconventional (such as the U-tail and the laminar flow wing).

- For engine bypass ratios above 9:1, FLOPS seemed to produce erroneous values for fuel consumption. Therefore, for all the concepts in the study using high bypass-ratio turbofans, the bypass-ratio was kept at 9:1, while the FLOPS fuel-flow calibration factor was adjusted such that the reduction in fuel consumption, compared with a current-technology baseline, matched published values (see the calibration section – Section 6.2.4.2). For the open-rotor (CROR) engine, a generic open-rotor engine deck, obtained from the Pacelab Aircraft Preliminary Design software [238], was imported into FLOPS and the fuel-flow calibration factor adjusted until it matched published expected fuel consumption reduction values for open-rotors (compared with a turbofan of equivalent technology – see Section 6.2.4.2). For future studies, the propulsion could be modelled using dedicated gas turbine performance simulation software, such as GasTurb [250] or NPSS (Numerical Propulsion System Analysis) [251]. However, for the purposes of this study, the above sufficed.
- The inbuilt noise module in FLOPS was also not entirely fit for purpose, as approach noise is not calculated and considerable adjustment/calibration would have been required to model the CROR noise. More advanced noise prediction methods are available (see for example [252]), but to keep the work within the overall scope of the research, a simple empirical model from Michel was adopted [249]. Although not developed for open-rotors, there are coefficients in this model that were adjusted such that the noise results of the CROR concepts matched published estimates (see the calibration Section 6.2.4.2). Again, this is crude, but at least allowed an ordinal comparison and therefore sufficed for the current research.

6.2.4.2 Workflow Calibration

To reasonably simulate the fuel-burn and noise emissions of the novel configurations, the workflow first had to be calibrated using data for a comparable existing

aircraft. Because the study was concerned with a possible replacement for an Airbus A320-type aircraft, this aircraft was selected as the baseline to calibrate the models with. After this initial calibration, selected parameters in FLOPS could be adjusted such that the reduction in fuel-consumption for the different engines (compared with the baseline) matched publicly available predictions. Similarly, the parameters in the noise model from Michel [249] was adjusted until the resulting noise emissions matched predicted values obtained from the literature. The calibration is discussed in detail in Appendix C.

6.2.4.3 Design-of-Experiments

To populate the evolvability design space, a full-factorial Design-of-Experiments (DoE) was performed for each CTC. The parameters varied, their associated minimum and maximum values, and number of levels are provided separately in Table 6.7 for combinations C1 and C3; Table 6.8 for C2a, C2b, C4a to d; and C5a to d, and Table 6.9 for C6a and C6b.

For all simulation runs, the number of passengers (150) and design range (2,700 nmi) was kept constant. The nacelle dimensions were also retained in each set of experiments (i.e. changing only across different types of engines), but the engine thrust was adjusted (by FLOPS) to maintain a selected take-off thrust-to-weight of about 0.308, which is a typical value for these types of aircraft (0.308 is the value for the A320 baseline). The engine masses were calculated by multiplying the FLOPS-internally-computed engine mass with the ratios of the new engine to CFM56 engine mass, as obtained from the calibration study. Some parameters (combined in the array ‘CT_parms’ – see Table 6.6) are dependent on the selected CTC (i.e. they change for each DoE set). These constitute the output of the ‘CT_select’ model (see Table 6.3)..

Table 6.7: Full-factorial sampling Design-of-Experiments set-up for concept-technology combinations C1 and C3.

	Min	Max	Levels
TR	0.2	0.3	3
SPAN [m]	30.5	33.5	5
AR	9	12	4
FlapChFr	0.25	0.275	2

The Design of Experiments for each CTC was set up and executed in AirCADia explorer. In total, the workflow was executed 4,960 times (240 times for 2015 C-T combinations and 4,720 times for those in the 2025-2035 timeframe) and the

Table 6.8: Full-factorial sampling Design-of-Experiments set-up for concept-technology combinations C2a to C5d (except C3).

	Min	Max	Levels
WF	3.95	4.15	2
TR	0.15	0.3	4
SPAN [m]	30.5	35	7
AR	11	14	4
flap_sw	1	3	2

Table 6.9: Full-factorial sampling Design-of-Experiments set-up for concept-technology combinations C6a and C6b.

	Min	Max	Levels
WF	3.95	4.15	2
SPAN [m]	30.5	36.6	6
AR	12	16	5
flap_sw	1	3	2

average execution time was less than two seconds on a standard desktop computer with an Intel Core i7-4770 CPU @ 3.40 GHz processor. Of these, 195 of the 2015 timeframe designs met the constraints for this scenario, whereas 1,444 designs met the constraints for at least one 2025-2035 timeframe scenario. These results were exported to MATLAB[®] scripts that produced the multi-attribute tradespace plots for the different scenarios. Because their technologies would not be available in 2015, the 2025-2035 era concepts did not produce utility values for the 2015 scenario and are therefore not available in this scenario. The results are provided, along with a discussion, in the next subsection.

6.2.5 Results and Discussion

The multi-attribute tradespaces (MATs) for the four scenarios, containing the results of the different configuration-technology combinations, are shown in Figure 6.3 and Figure 6.4. In these figures, each marker (point) represents an individual aircraft design, each colour represents a configuration, and the different marker symbols correspond to the different technology combinations within the separate configurations. The spread within each CTC set of data points reflect the different levels of the parameters that were varied in the design of experiments. The cost shown is the total development (non-recurring) cost value per aircraft in 2017 US

dollars (a production run of 1,000 units was assumed).

Keeping the assumptions and limitations of the computational models in mind, it can be observed from the BBT 2015 scenario that the conventional tube-and-wing is generally (depending on the airframe dimensions) less expensive to develop than the T-tail configuration. This is because of the higher fuselage and wing mass that FLOPS assign to T-tails, which also leads to an increase in the mass of other components, and subsequently higher cost, as the cost per unit-mass is constant for a selected airframe component. The higher mass also detrimentally affects utility, because more fuel is required to transport the heavier aircraft and the higher-thrust engines and heavier airframes emit more noise. The EIS 2015 designs can also be seen at the bottom of the other scenarios where their ‘dated’ technology makes them less desirable.

As can be seen in Figures 6.3 and 6.4, for the 2025-2035 EIS period the ranking of concepts, based on utility, is different in each scenario. In KC 2035, designs incorporating open-rotors and laminar flow wings dominate if cost is not considered. This is expected, as the CROC engines and laminar flow wings are the technologies that would most radically reduce fuel-consumption (of the technologies modelled), which is the main consideration in the KC scenario.

A different situation can be observed in NiMBY, where, as would be expected, the noise-shielding U-tails dominate in terms of utility. The noisy open-rotors now perform the worst, apart from the designs from the EIS 2015 period.

In the BBT 2035 scenario, the disparity in the desirability of the different engines become less distinct. However, passive laminar flow wings have lower utility, as these cruise at lower speeds, where, in BBT 2035, higher speeds are desired. Although not visible from the tradespace, as would be expected, the fuselage width is higher for the higher utility designs and all the designs have larger wing areas compared with the other scenarios. Most of the designs also use the triple-slotted flap. This is because these are the designs that met the strict field-length requirements of this scenario. It should be noted that the use of triple-slotted flaps increases mass and complexity, and subsequently incurs additional manufacturing and maintenance expenses. This higher mass and cost were taken into account in the example by using methods from Rudolph [221]. In reality, the disadvantages associated with triple-slotted flaps will probably result in potentially very tight future field-length requirements being challenged, or other means for improving field performance being sought. In Figure 6.4, the designs of zero utility (at the bottom of BBT 2035) are designs that did not meet the field-length constraint for BBT 2035, but which are still available in the other scenarios.

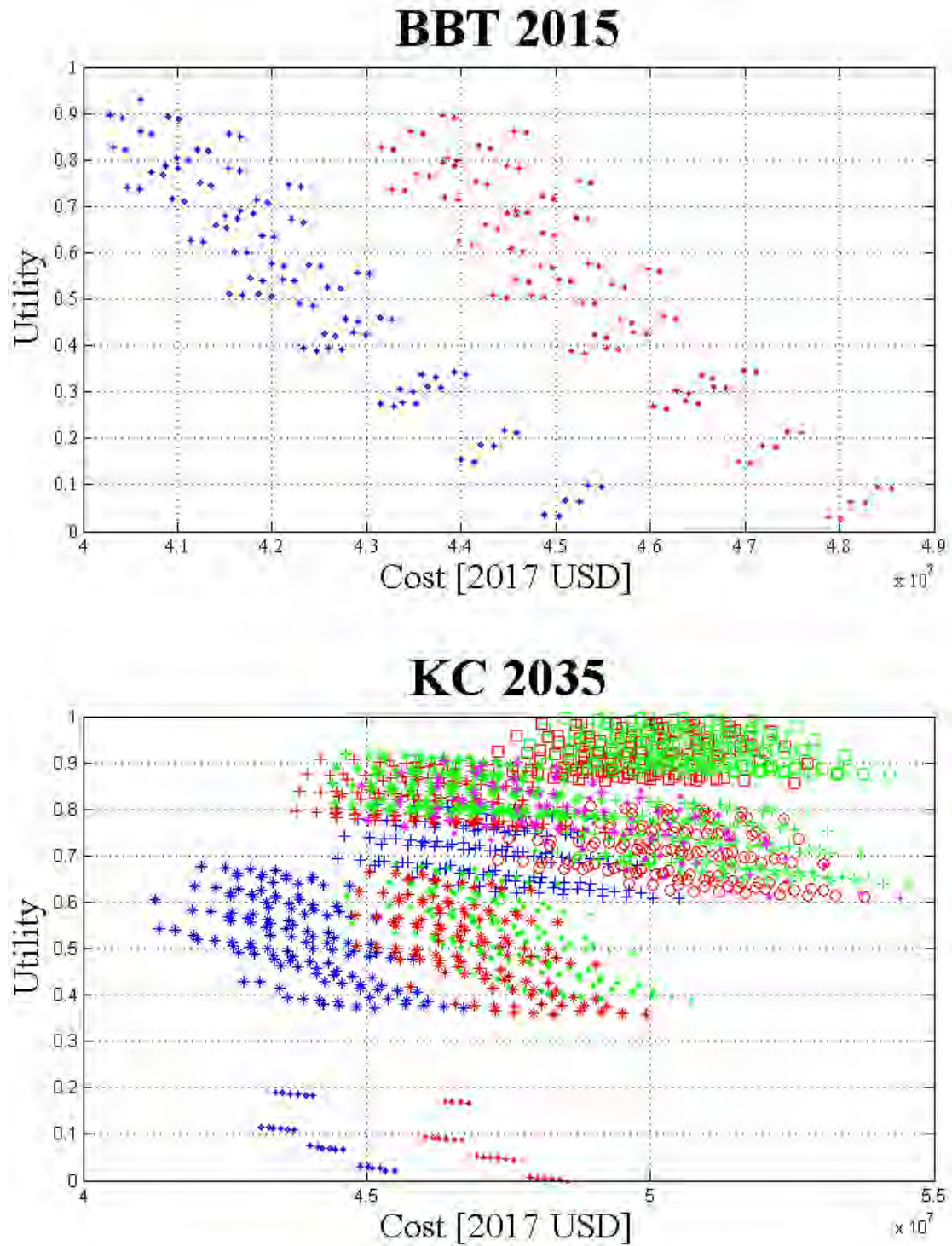


Figure 6.3: Multi-Attribute Tradespace results of the full-factorial sampling experiments for scenarios BBT 2015 and KC 2035. Each point represents a single design, whereas the colours and symbols each represent a different configuration (see Figure 6.1).

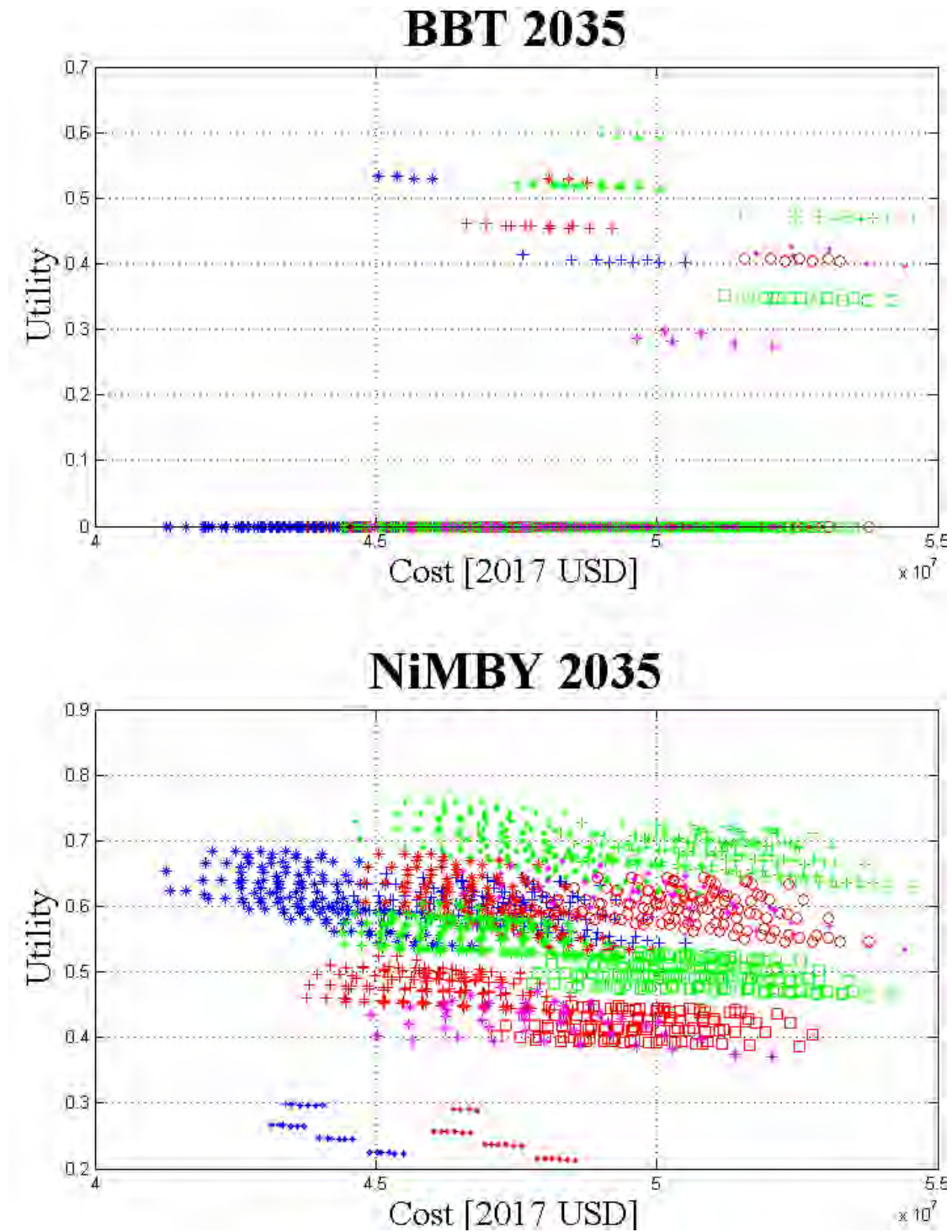


Figure 6.4: Multi-Attribute Tradespace results of the full-factorial sampling experiments for scenarios BBT 2035 and NiMBY 2035. Each point represents a single design, whereas the colours and symbols each represent a different configuration (see Figure 6.1).

Although never having the highest utilities, aircraft of the conventional tube-and-wing layout perform reasonably well across the scenarios, especially in NiMBY 2035 and BBT 2035. It also remains the least expensive option (except where laminar flow is incorporated).

Finally, it can be seen in the two figures that the 2015 technology aircraft have the lowest utilities in the 2025-2035 scenarios. They still exhibit high cost, because the cost is computed as if the designs are clean sheet developments.

With the evolvability design space now populated with example designs, the similarity assessment and set-based design techniques could be applied to these for demonstration and evaluation purposes. This is the subject of the next subsection.

6.3 Airframe Similarity Assessment Framework Evaluation

6.3.1 Purpose

The purpose of the work presented in this section was twofold:

1. To test the airframe similarity assessment framework, constituting the commonality prediction techniques proposed in Chapter 4, (Sections 4.4 and 4.5). Testing here refers to determining whether the techniques function as required and provide expected results. To determine whether the techniques function correctly, it had to be shown that they could accurately identify airframe components (or segments of these) that are known to be common across existing aircraft designs. Furthermore, this also had to be demonstrated across designs that have radically dissimilar configurations. To demonstrate that the techniques provide expected results, the results had to be compared with publicly available descriptions of commonality across actual aircraft (see Chapter 3 for this).
2. To demonstrate how the commonality prediction techniques would be exploited in a conceptual design evolvability exploration study, using the demonstration case study presented in the previous section (Section 6.2).

To be able to perform this evaluation, the commonality prediction techniques introduced in Chapter 4 first had to be implemented into a software application. This is the subject of Section 6.3.2 and is followed by Section 6.3.3, which covers the application and testing of the techniques using information of actual Airbus and Boeing aeroplanes. In Section 6.3.4, the testing of the techniques on aircraft with novel and dissimilar configurations is described, followed in Section 6.3.5, with

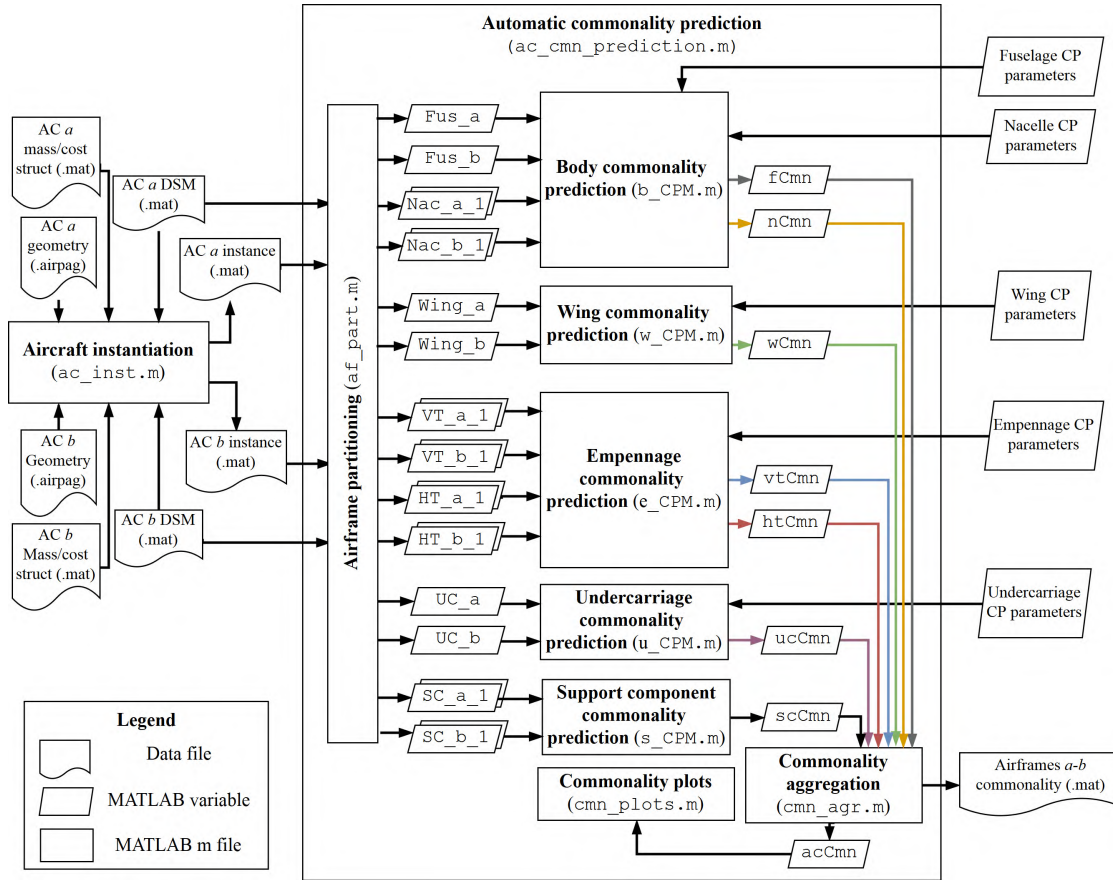


Figure 6.5: Automatic commonality prediction computational workflow.

demonstrations of how the techniques would be employed using the demonstration case study. The computational cost associated with the techniques are discussed in Section 6.3.6.

6.3.2 Software Implementation

The commonality prediction techniques and algorithms proposed in Chapter 4 were implemented into a suite of MATLAB[®] scripts and functions. A much-simplified overview of the software implementation is presented in Figure 6.5, which illustrates the computational workflow.

As can be seen in Figure 6.5, the process initiates with data files containing information on the two selected aircraft (*a* and *b*) being fed to the aircraft instantiation module (implemented in the `ac_inst.m` file). This module produces the instantiated objects for aircraft *a* and *b* as ‘.mat’ data struct files (please refer to Section 4.3.1 for a reminder of the structure of the object model). These are then input to a large m file (`ac_cmn_detector.m`), which calls several other m-files as function. The first of these is the airframe partitioning module (`af_part.m`), which breaks up the

aircraft into their constituent components and decides which components should be sent for detailed comparison. The output of the partitioning are the pairs of components that have to be compared (note that, for certain components, such as the nacelles and empennage surfaces, more than one pair could be output – for this research it was assumed that each configuration would only have a single fuselage and a single wing). These component pairs are embodied in struct variables that contain all the necessary parametric, mass and cost information to perform the commonality prediction.

The structs are then inserted to the different commonality prediction modules, which are functions called by `ac_cmn_detector.m`. These produce commonality structs as output, which contain data on the common mass of the components and the associated cost reduction. These are fed to the commonality aggregation function, which calculates the overall airframe commonality and saves it in the ‘acCmn’ struct. This struct is saved as a `.mat` file and also inserted to a commonality plotting function, which produces geometry renderings showing the common components across the two aircraft.

More details on each of these modules, along with their respective input and output data, are provided in Appendix C.3.

6.3.3 Application to Existing Aircraft

The first part of the automatic airframe similarity assessment framework evaluation involved applying the techniques to actual existing aircraft baseline-descendant pairs. The purpose was to qualitatively compare the commonality results produced by the proposed methods with published descriptions of commonality achieved in real applications. The approach followed is discussed in the subsection below, followed by the results and discussion.

6.3.3.1 Approach

The baseline-descendant aircraft pairs selected for this evaluation are listed in Table 6.10. The rationale for their selection is also listed, along with descriptions of how the mass breakdown was obtained/calculated. All geometry information was obtained from the manufacturers’ three-view CAD drawings ([224] for Airbus and [151] for Boeing), except for the Boeing 707, where the airport planning manual [240] was used to obtain this information. Where not clear from the CAD drawings, wing spar locations were obtained from Sensmeier and Samareh [253, p. 3].

With the available geometry and mass information, the workflow described in Section 6.3 was executed for each baseline-descendant pair. The output geometry

Table 6.10: Existing baseline-descendant (BL-D) aircraft pairs selected.

OEM ^a	BL	D	Reason for selection	Mass calculation
Airbus	A320	A320	To determine if techniques can detect that same aircraft is used. To check for CP ^b errors.	Mass breakdown obtained from default A320 in PLAPD ^c .
Airbus	A320	A321	To test CP for simple stretches and different trailing edge flaps.	Mass breakdowns obtained from default A320 and A321 in PLAPD.
Airbus	A340-300	A330-300	To test CP for different engine configurations and undercarriage layout.	Mass breakdowns obtained from default A340-300 and A330-300 in PLAPD.
Airbus	A340-300	A340-600	To test CP for large wing chord increase and different undercarriage layout.	Mass breakdowns obtained from default -300 and -600 in PLAPD.
Boeing	707-320C	727-200	To test CP for double bubble fuselages, and different empennage and engine arrangements.	Sized with NeoCASS [239] (calibrated with the mass data in Ref. [26, Data Set A]).
Boeing	727-200	737-200	To test CP for double bubble fuselages, and different empennage and engine arrangements.	Sized with NeoCASS [239] (calibrated with the mass data in Ref. [26, Data Set A]).
Boeing	737-200	737-400	To test CP for slat and leading-edge extension changes.	Sized with NeoCASS [239] (calibrated with the mass data in Ref. [26, Data Set A]).

^a Original Equipment Manufacturer.^b Commonality prediction.^c PACELAB APD (Aircraft Preliminary Design) software [238].

renderings were then compared with published information. This information is qualitative in nature and is discussed at length in Chapter 3 for all the aircraft considered here. Meaningful commonality information for commercial aircraft are not published and it is therefore not possible to compare the output commonality scores with actual data. However, the scores were shown to aircraft design specialists at Airbus UK (see Section 6.5) to determine whether they are reasonable (i.e. match expectations for the given aircraft, while remaining cognisant of the assumptions involved) and would at least aid in making ordinal comparisons across different baseline-descendant pairs.

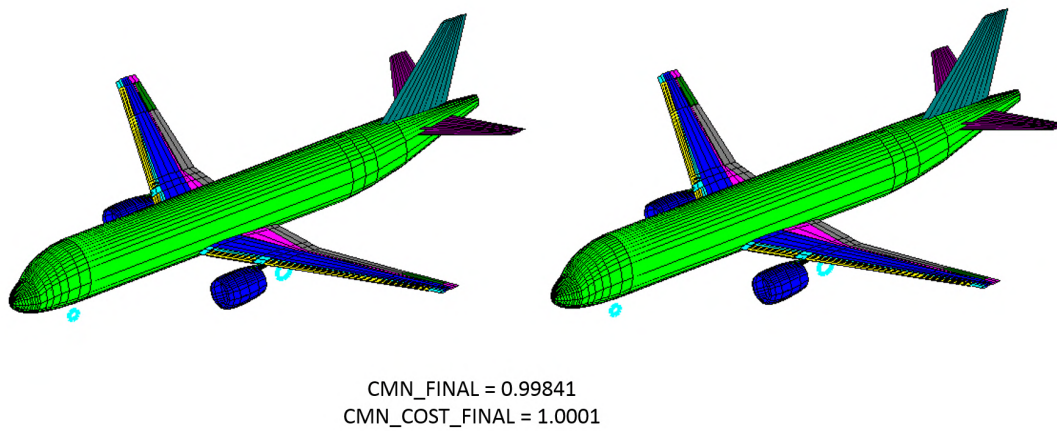


Figure 6.6: Commonality prediction for the Airbus A320 and A320 ('sanity check').

6.3.3.2 Results and Discussion

The results of the commonality prediction for existing aircraft baseline-descendant pairs are shown in Figures 6.6 to 6.12. In these figures, components/segments that are rendered in white (or black for the undercarriage) show that the particular component or segment is not common with the other aircraft. The coloured components/segments show that these are common with a component/segment of the other aircraft, which is rendered in the same colour and is of the same type and relative location). Also shown are the overall mass and cost commonality scores produced by the workflow presented in Section 6.3. These are called 'CMN_FINAL' and 'CMN_COST_FINAL', respectively.

Figure 6.6 shows the commonality prediction results when the A320 is inserted to the workflow as both a baseline and descendant. The reason for doing this is to see whether the techniques and software implementation could detect that the aircraft are 100% common to each other. Another reason is to see whether there are any major numerical errors that arise (which, if present, would become apparent in the bodies and wing). This comparison is therefore a 'sanity check'. As can be seen from Figure 6.6, the software correctly determined that the aircraft are exactly the same. The slight error in the commonality scores are due to the discretisation and are negligibly small.

The A320/A321 comparison is shown in Figure 6.7. Here, it can be seen that it was correctly determined that the A321 fuselage has two plugs. However, note that in the actual A321, the rear plug is actually just behind the wing, not as far back as shown. The reason this happens is that the body longest common subsequence solver has no parameter which it can use to decide where the non-common segment

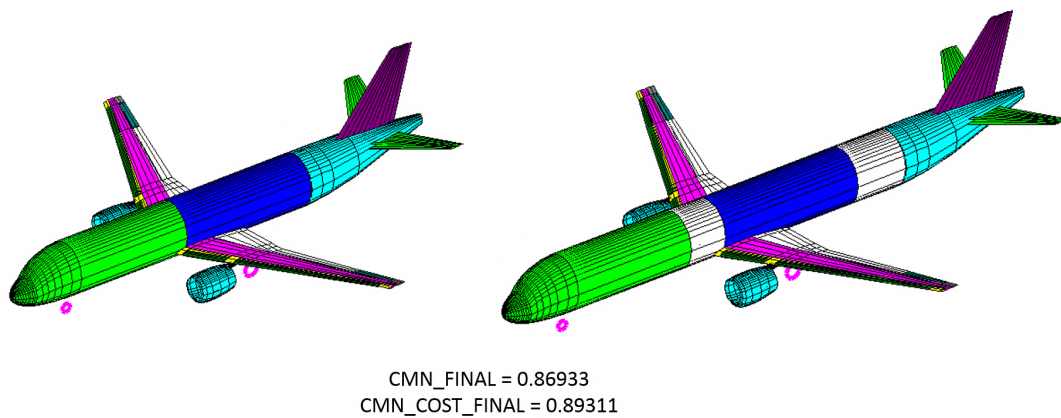


Figure 6.7: Commonality prediction for the Airbus A320 and A321.

of two tubular body segments with the same, constant cross-section distributions should be. The solver will simply assign the equal sub-segments it can find to be common until it reaches the end of the shorter segment. This is not a particular problem when calculating the commonality scores, except when there is a large gradient in the mass distributions along these segments. For future work, it is suggested that rules be implemented to correctly assign the non-common segments for two constant cross-section tubular segments. The cost commonality score is slightly higher than the mass score. This is because the more expensive, but lighter parts of the fuselage remain common.

As can also be observed from Figure 6.7, the fixed trailing-edge and trailing-edge flaps were assigned to be non-common across the two aircraft. This assignment is correct, as the A321 employs slightly larger double-slotted flaps than the single-slotted flaps of the A320 (see Chapter 3.4.3). However, on the actual A321, there might be some commonality with the A320 for the fixed trailing-edge. For the current implementation, it was assumed that, if the flap type is different across the two aircraft being compared, the fixed trailing-edge would be non-common. More information regarding the actual commonality achieved will be needed to accurately assign commonality using the techniques presented here. To find such information was, however, beyond the scope of this research.

The commonality prediction for the Airbus A340-300 and A330-300 is shown in Figure 6.8. As can be seen, all airframe components were assigned to be common, apart from the nacelles, the centre main undercarriage leg and wheels of the A340, and the fixed leading-edge and slat at the location of the outboard engine of the A340. This matches the description of the commonality across these two aircraft (see Chapter 3.4.4). The cost commonality score is again higher than the mass score. This is because the non-common components are relatively less expensive

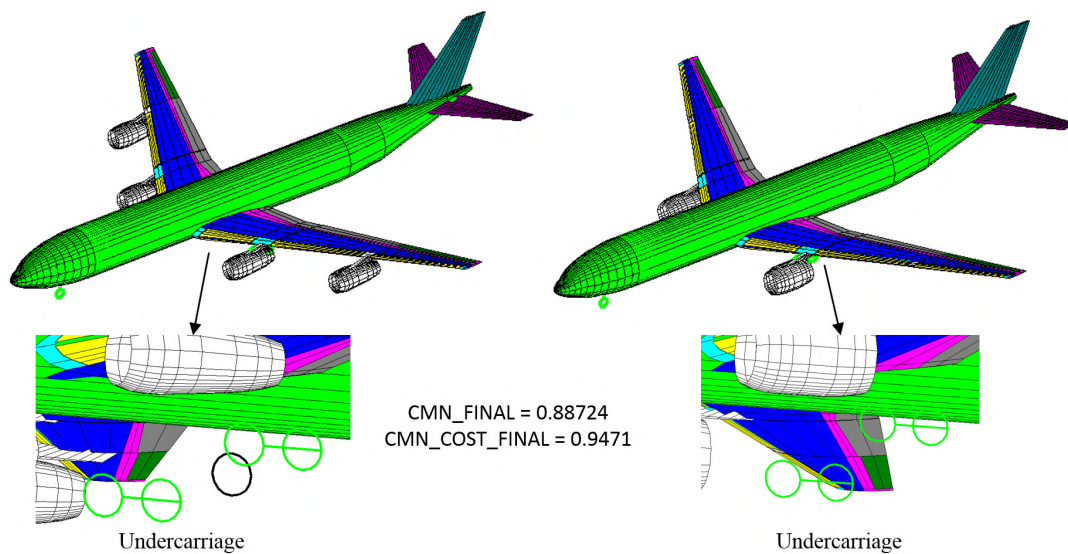


Figure 6.8: Commonality prediction for the Airbus A340-300 and A330-300.

than the common ones (as determined by the cost model used [104]).

The more extreme modification of the A340-300 to devise the A340-600 are clearly prevalent in Figure 6.9. The nacelles and centre main undercarriage leg are correctly assigned to be non-common. The wedge-shaped chord-extending insert discussed in Chapter 3.4.4.2 can also be seen. Note that the rendering places chord-inserts in the middle of the space between the spars, and the location shown is therefore not necessarily the actual location of the insert. It can also be observed that the wingspan extensions were correctly identified as tip extensions. The tail surfaces are shown to be common apart from root inserts. The A340-600 indeed received a vertical tail extension, but the horizontal tail was actually ‘photographically’ scaled. The commonality prediction for empennage surfaces does not determine if span extensions or scaling are used (it only uses the difference in surface area and mass to compute commonality), but the geometry rendering was coded to always show a root-insert or tip-extension. This is deemed acceptable, as the exact empennage planform shape is usually not of considerable importance during conceptual design (as the surface area is). Finally, it can be seen that the commonality scores computed are considerably lower than for the A340-300 and A330-300 pair.

Figures 6.10 and 6.11 show the commonality predictions for the Boeing 707-320C/727-200 and 727-200/737-200 pairs. These tests were performed as an initial check to determine if the software implementation could manage aircraft with different configurations and double-bubble fuselages. In the cases shown, the only

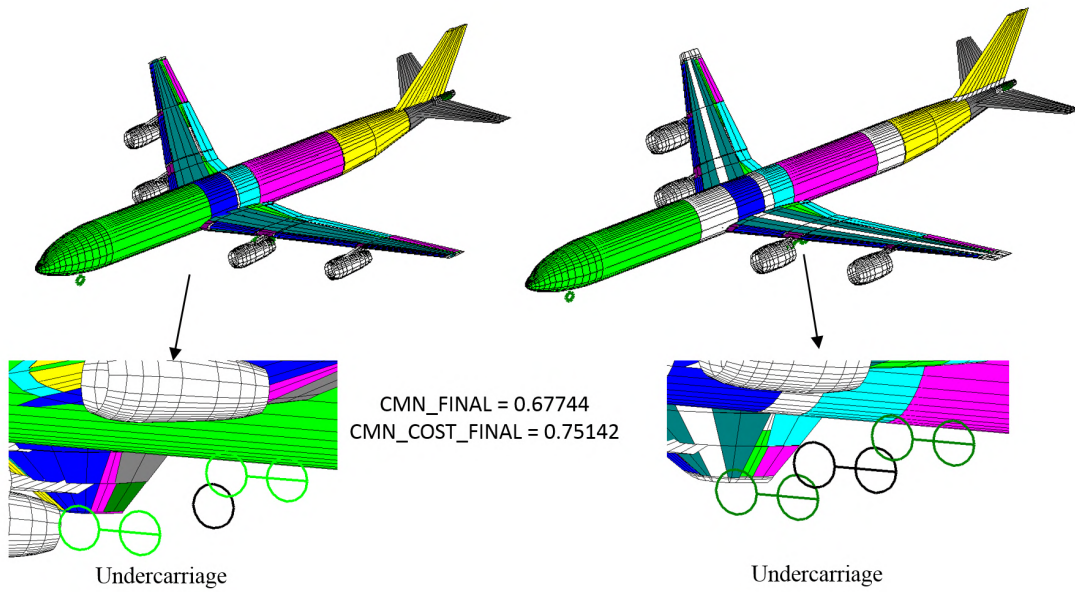


Figure 6.9: Commonality prediction for the Airbus A340-300 and A340-600.

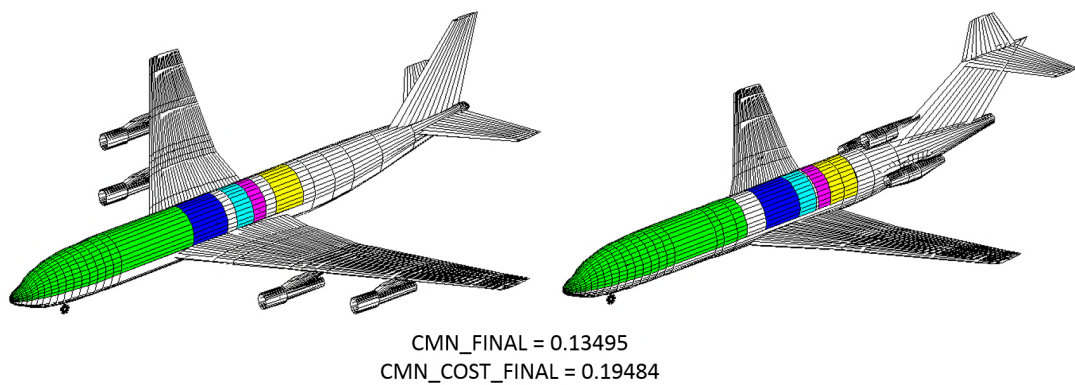


Figure 6.10: Commonality prediction for the Boeing 707-320C and 727-200.

commonality found was across the fuselages. There are common segments across the top lobes of the 707 and 727 and across the 727 and 737 noses and segments of the top lobes of the fuselages further aft. This reuse of the top lobe (and nose for the 727/737) is indeed what was done for these aircraft (see Sections 3.2.4 and 3.2.5). As would be expected, the commonality scores for these examples are considerably lower than for the previous cases.

The changes from the original Boeing 737s (the -200) to the Classics can be seen in Figure 6.12. The reader can compare the rendering on the right-hand side (the 737-400 Classic) in this figure with the 737-300 shown in Figure 3.4. As can be

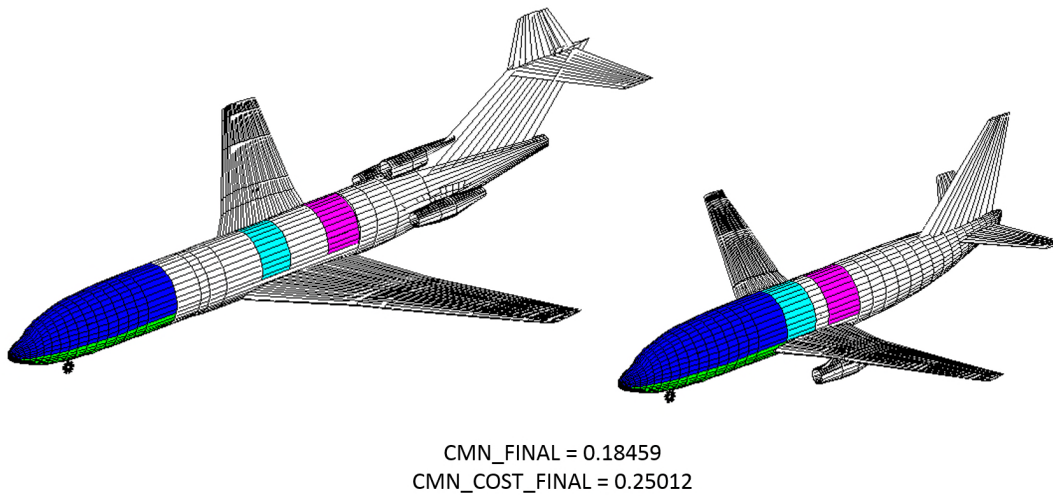


Figure 6.11: Commonality prediction for the Boeing 727-200 and 737-200.

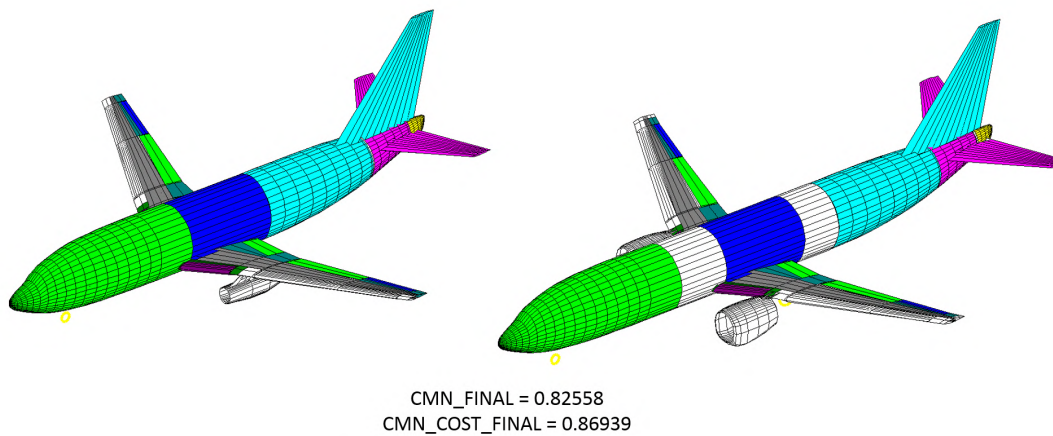


Figure 6.12: Commonality prediction for the Boeing 737-200 and 737-400.

seen, the fixed leading-edge and outboard slat were correctly assigned to be non-common. The tip-extensions on the -400 are also visible, as are the new nacelles. The plugs are longer than those shown in Figure 3.4, as the 737-400 has a longer fuselage than the -300.

The only differences between the results in Figure 6.12 and the changes shown in Figure 3.4 are in the empennage surfaces and the region of the wing where the engines are located. As already mentioned, the empennage commonality prediction module only employs surface area and mass when computing commonality and does not determine whether the changes are through scaling, a tip-extension, or a root-insert (or wedge as shown for the -300). The rendering did assign root-inserts for both the vertical and horizontal tails for the -400, but these are not readily visible

in the figure, as they are small and partially located inside the fuselage rendering. Regarding the difference at the engine location, the wing commonality prediction does not consider the location of fuel tanks, and the difference is therefore not picked up. This is a feature that could be easily added as part of future work.

The results in this section showed that the proposed commonality prediction techniques could accurately determine which airframe components or segments of these were reused across actual baseline-descendant aircraft pairs. The next step was to determine how well the techniques and the software could manage pairs of aircraft having considerably different configurations. This is the subject of the next section.

6.3.4 Application to Dissimilar Configurations

The tests presented in this section were devised to demonstrate that the proposed techniques and software implementation could correctly handle substantially dissimilar configurations. The approach is discussed next, followed by the results, along with a discussion.

6.3.4.1 Approach

For this test, a conventional baseline configuration was first established. The mass properties and geometry (apart from a more modern nose) of the A330 were directly applied to this design. Next, three novel and unconventional configurations were modelled. The objective was to produce a selection of considerably different configurations. Based on this, three radically different concepts were selected: a boundary-layer ingestion propulsive fuselage concept (such as proposed in Seitz et al. [254]), an aircraft similar to the ‘Airbus Concept Plane’ (see Airbus [255]), and a generic blended-wing-body. These will be referred to as ‘BLI’, ‘ACP’, and ‘BWB’ respectively.

To accurately model and simulate these configurations was outside the scope of this research. Significant assumptions were therefore made regarding the modelling of the geometry and the mass breakdowns of the airframe components. The geometries were created to roughly match images for these aircraft. The nose and wing (excluding the centre section) of the baseline were essentially reused as-is on all the concepts. The DSMs were created manually. The conventional undercarriage was also reused on the ‘BLI’ and ‘ACP’ concepts. The same airframe mass breakdown for the conventional configuration was used for all the concepts (but distributed differently according to the separate cross-section distributions). For the ‘BLI’ concept, 1,000 kg was added to the rear fuselage to account for the

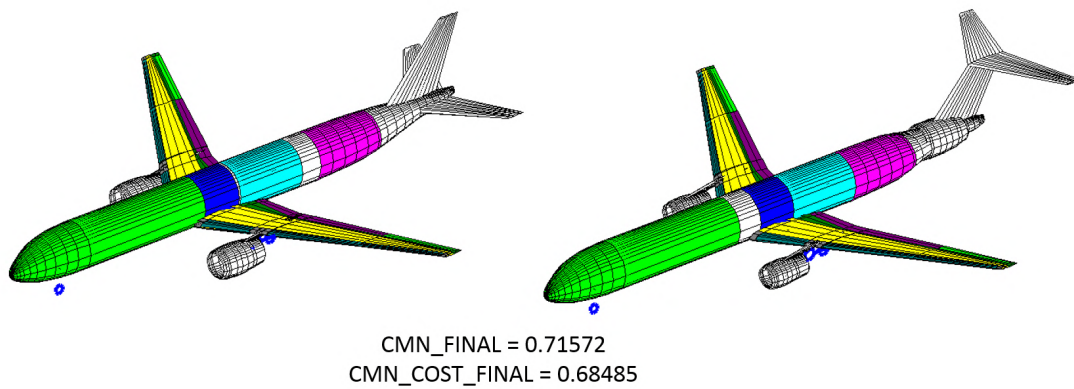


Figure 6.13: Commonality prediction for a conventional configuration and a boundary-layer ingestion propulsive fuselage concept.

structural reinforcements needed for the T-tail and engine housing¹. The cost of new segments/components was multiplied by a factor of 1.5.

These simplifications were deemed reasonable, because the purpose of this study was solely to see how well substantially different configurations are handled by the proposed techniques. This could be confirmed visually, based on the output geometry renderings. However, the commonality scores might be inaccurate and should be interpreted with the above assumptions in mind.

6.3.4.2 Results and Discussion

The results are shown in Figures 6.13 to 6.15. As can be seen in Figure 6.13, which shows the commonality prediction for the conventional and BLI concept-pair, the airframe partitioning module correctly assigned the conventional and T-tail empennages to not be compared. The rear fuselages were also correctly determined to be non-common, because there is an engine in the one and not the other. There are also non-common fuselage plugs that account for the different position of the wing. The other components were correctly assigned to be common (or non-common for the nacelles).

Figure 6.14 shows the commonality prediction for the conventional and ACP concept-pair only the noses are common across the two fuselages. As expected, the empennages and nacelles were determined to be non-common. The rear part of the inboard wing was also correctly determined to be non-common, as there is a nacelle attached there for the ‘ACP’. Similarly, the wings are non-common at the location of the engine of the baseline configuration.

¹However, it should be noted that the fuel-burn savings associated with the BLI concept will likely lower overall airframe mass.

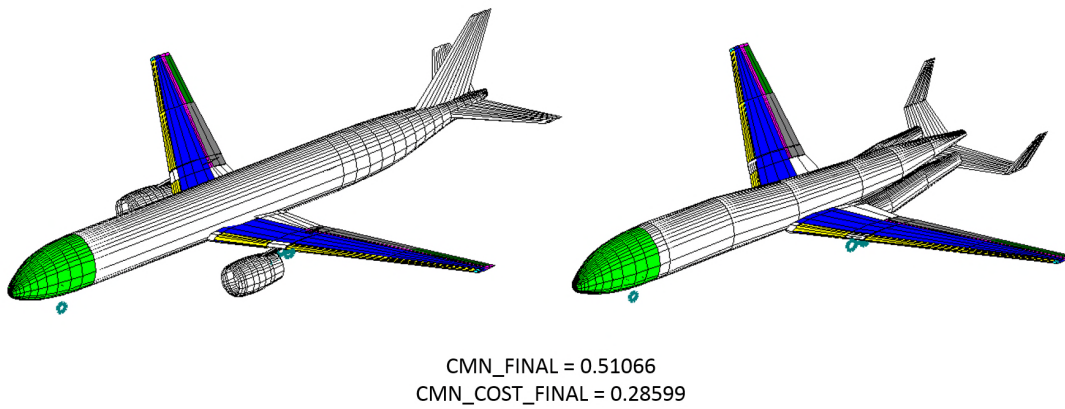


Figure 6.14: Commonality prediction for a conventional configuration and the Airbus future concept.

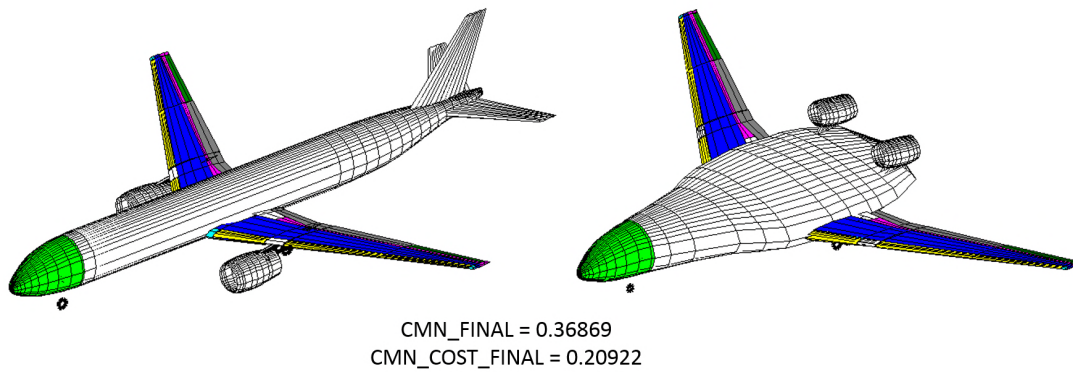


Figure 6.15: Commonality prediction for a conventional configuration and a blended-wing-body concept.

In the final case, that of the conventional and BWB pair (Figure 6.15), the fuselage is determined to be mostly non-common (apart from the nose). The undercarriage is also non-common, because, for the BWB, the main undercarriage is attached to the fuselage and not the wing. Finally, the empennage of the conventional aircraft is correctly determined to be non-common, as the BWB does not have one.

These results showed that the software implementation (especially the airframe partitioning module) worked as expected and that the commonality prediction techniques could correctly handle airframe components connected in different ways across a baseline-descendant pair.

6.3.5 Application to Demonstration Case Study

In this section, the commonality prediction was executed for the aircraft designs in the demonstration case study introduced in Section 6.2. The purpose was twofold:

i) to further test the techniques for dissimilar configurations, and ii) to demonstrate how the techniques could be used for evolvability studies during conceptual design. The approach followed is discussed next, followed by the results and a short discussion.

6.3.5.1 Approach

Only the top 25 designs from both C1 and C3 were considered as baselines. These were selected by adjusting a Fuzzy-Pareto number [14] for each set (C1 and C3), until 25 designs remained in the Fuzzy-Pareto set of each. All the designs in the 2025-2035 timeframe were considered potential descendant aircraft. The commonality prediction workflow in Section 6.3.2 was therefore executed for each pairwise combination of 2015 and 2025-2035 aircraft. The output of the design space sampling in Section 6.2.4.3 was ready to be fed to the workflow, which was run in batch mode.

6.3.5.2 Results and Discussion

The workflow was executed a total number of 72,200 times (50 baselines times 1,444 possible descendants). The average computational time was calculated to be 2.0768s on a desktop PC with an Intel Core i7-4770 CPU @ 3.40 GHz processor. The computational cost is discussed in more detail in Section 6.3.6.

The commonality results obtained from the batch run could be used in several different ways to support evolvability-related decision-making. One way is to apply the proposed set-based design techniques (which are demonstrated in Section 6.4) to filter out designs that are dominated by more desirable designs. Here, however, it will be shown how the commonality results could be exploited to perform interactive evolvability exploration. Note that these interactive techniques were not implemented in a software application, but doing so should be straightforward, especially using a software tool such as AirCADia Explorer.

To create the plots that follow, the commonality results were inserted into ‘baseline-descendant commonality evolvability exploration matrices’ (see Figure 6.16). These matrices contain the cost or mass commonality scores between the baselines (columns) and descendants (rows), or the cost values of the descendants with respect to the baselines. The different exploration plots are created when filtering these matrices, as will be discussed next.

Figures 6.3 and 6.4 showed the utility and cost results of the demonstration case designs before any similarity assessment prediction has been performed. Using the

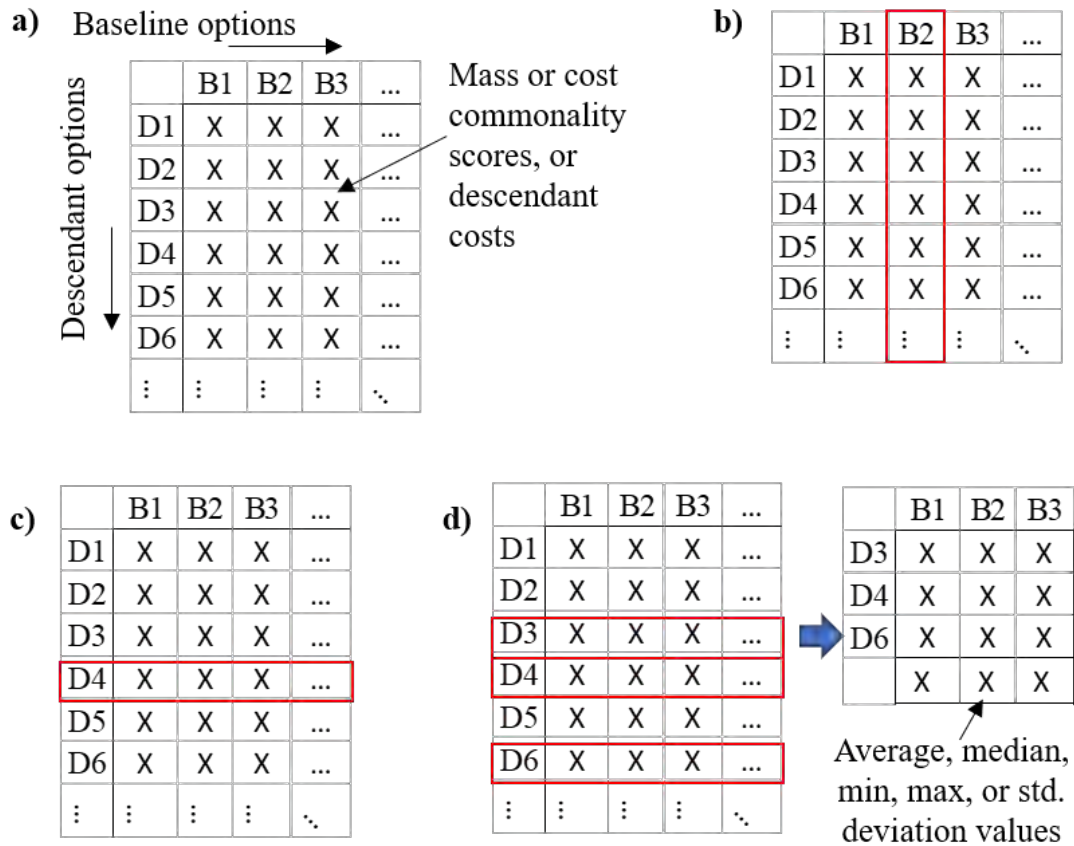


Figure 6.16: Exploring evolvability options. a) The baseline-descendant commonality evolvability exploration matrix. b) Past-to-future exploration (selecting a baseline and exploring the commonality/cost of descendant options for this selected baseline). c) Future-to-past exploration for a single selected descendant (selecting a descendant option and exploring the commonality with baseline options for this selected descendant). d) Future-to-past exploration for a multiple selected descendants .

commonality results obtained here for these designs, the user could select a particular baseline of interest in the 2015 timeframe. To do this, the baseline-descendant cost matrix with descendant cost values is used, which is filtered by extracting a column from the matrix (see Figure 6.16b), i.e. for a selected baseline, all the cost values of the descendants are extracted to create the plots. The cost values of the far-future designs are subsequently updated to reflect the cost reduction due to commonality computed with the selected baseline design. This is illustrated in the following figures.

Figures 6.17 and 6.18 illustrate how the cost of the different far-future designs in the evolvability demonstration study change if a conventional configuration is selected in BBT 2015. As can be seen from the figure, commonality with the baseline has resulted in the cost of many of the designs to be lower than if they were to be produced as clean-sheet designs (as shown in Figures 6.3 and 6.4). As would

be expected, the cost of the far-future designs with the same configuration (C2a and C2b [see Figure 6.1]) are reduced more than the those of other configurations. However, these are not the best performing designs (in terms of utility). The highest utility designs with the lowest commonality-adjusted development cost are the U-tails and T-tail configurations (without laminar flow – i.e. C4a, C4b, C5a, and C5b [see Figure 6.1]). These have somewhat similar commonality benefits to each other but dominate in different far-future scenarios.

Example high-utility and/or low-cost possible descendants of B1, namely D1, D2, and D3 are highlighted in Figures 6.17 and 6.18. Geometry renderings of these descendants, displaying components with predicted commonality, are shown in Figure 6.19, along with the associated mass and cost commonality scores.

The situation changes drastically when a baseline with a T-tail configuration is selected in BBT 2015. This is illustrated in Figures 6.20 and 6.21. As can be seen, the far-future designs with a T-tail layout (except for laminar flow wings) now exhibit more commonality cost-reduction benefits than any other configurations. These generally also have higher utility in most of the scenarios as compared with the conventional baseline.

As with the conventional baseline, three example possible descendants of T-tail B1, namely D1, D2, and D3, are highlighted in Figures 6.20 and 6.21 (note that these are different examples than those in Figures 6.17, 6.18, and 6.19). Geometry renderings with predicted commonality for these are displayed in Figure 6.22.

The commonality scores for a specific selected baseline could also be visualised such as in Figure 6.23, in which the colours correspond to the degree of cost commonality with the selected baseline.

Another exploration technique that the techniques enable is to select a desirable far-future design and view which near-future designs have the commonality with this selected design (see Figure 6.16c). An example of such a case can be seen in Figure 6.24, where a single design was selected for the far-future (highlighted as a black point in the 2025-2035 scenarios). The colours in the 2015 scenarios correspond to the commonality of the 2015 designs with the selected far-future design.

The above illustrates how a full software implementation could be exploited to allow interactive exploration. In such an implementation, it is envisioned that the designer would click on different designs options and immediately obtain feedback regarding the predicted commonality with other near- or far-future design options. If designs in both the near- and far-future timeframes are selected he/she could investigate the commonality results in detail by inspecting the geometry renderings.

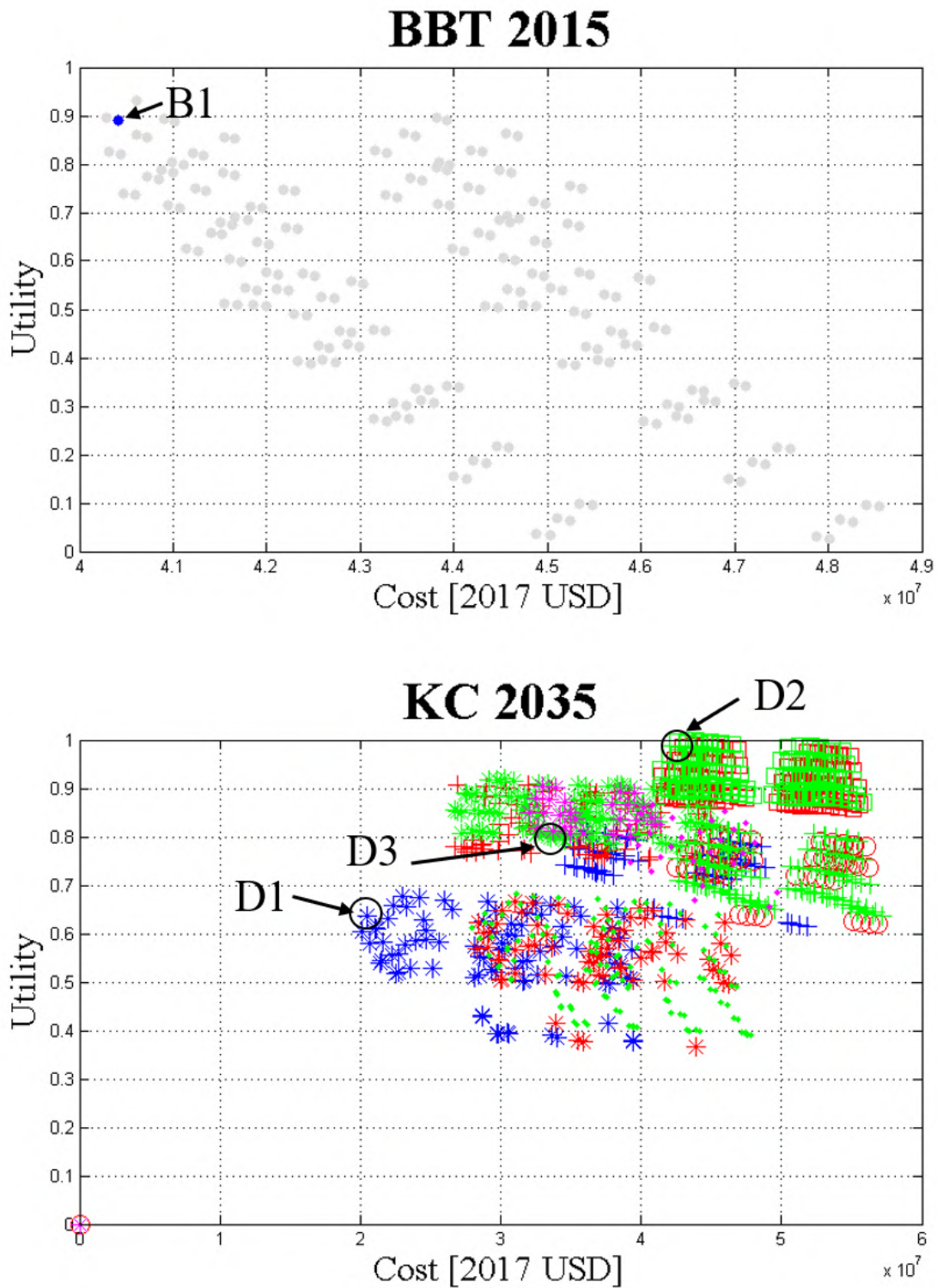


Figure 6.17: Investigating descendant cost in far-future scenario KC 2035, for a selected conventional baseline (B1) in BBT 2015. Each point represents a single design, whereas the colours and symbols each represent a different configuration (see Figure 6.1 for a guide). Geometry renderings, displaying the predicted commonality between design option B1 and example possible descendants D1, D2, and D3, are shown in Figure 6.19.

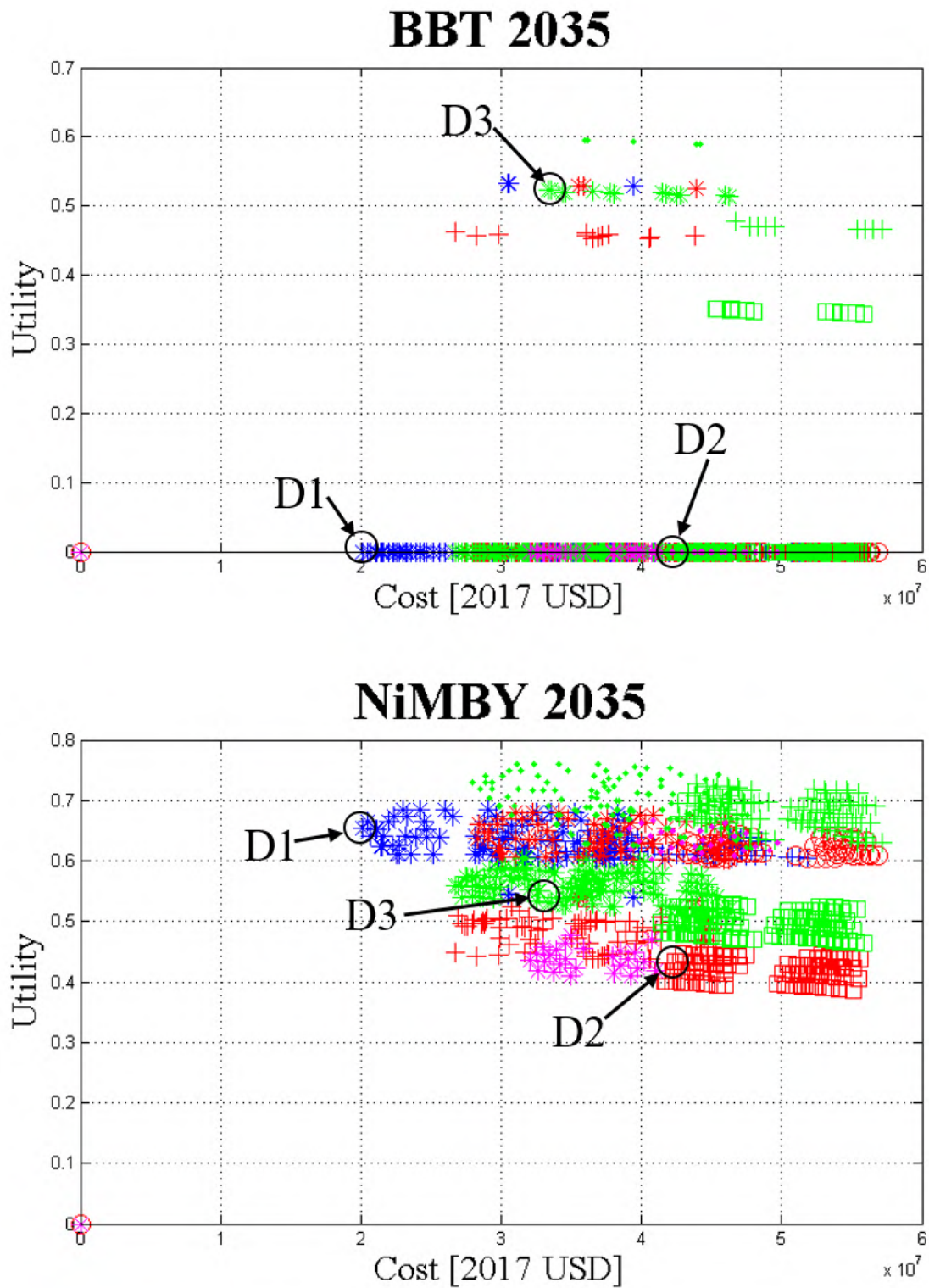


Figure 6.18: Investigating descendant cost in far-future scenarios BBT 2035 and NiMBY 2035, for a selected conventional baseline (B1 in BBT 2015 in Figure 6.17). Each point represents a single design, whereas the colours and symbols each represent a different configuration (see Figure 6.1 for a guide). Geometry renderings, displaying the predicted commonality between design option B1 and example possible descendants D1, D2, and D3, are shown in Figure 6.19.

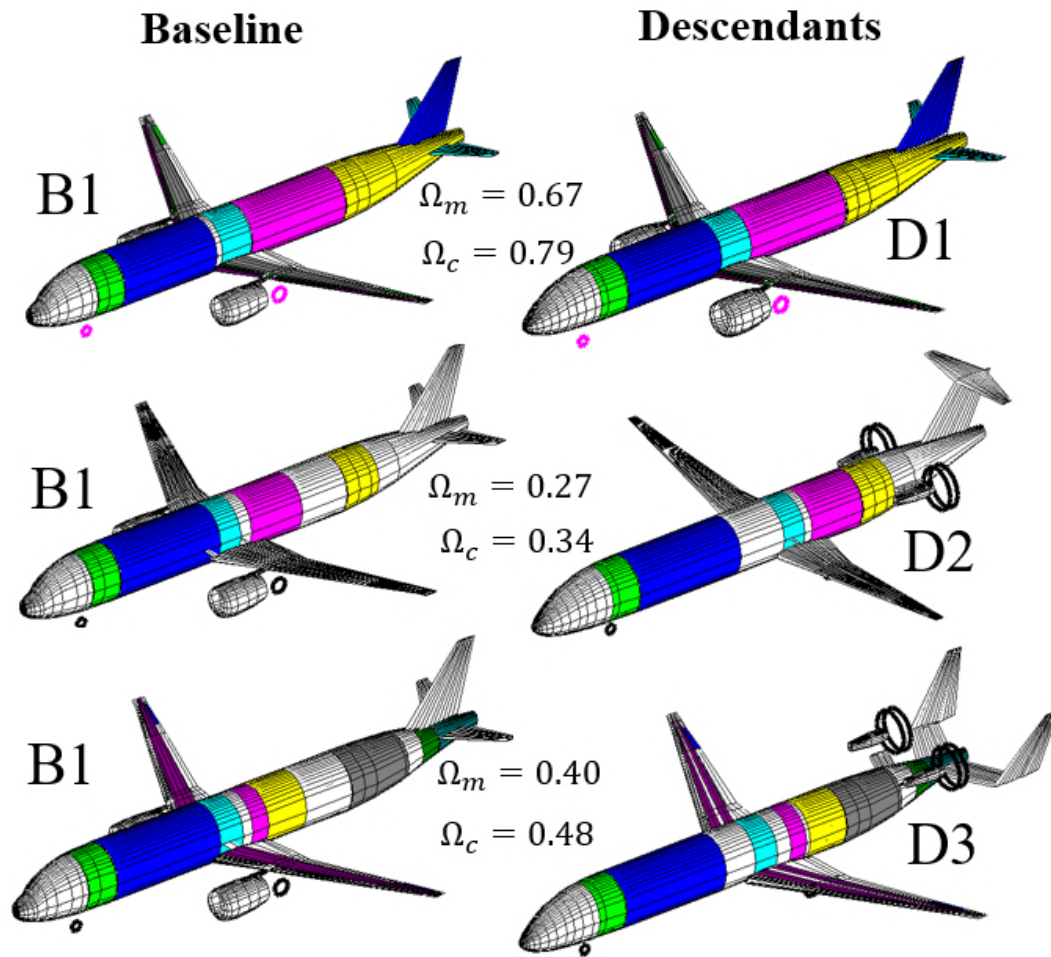


Figure 6.19: Geometry renderings, displaying the predicted commonality between design option B1 and example descendant options D1, D2, and D3 in Figures 6.17 and 6.18.

The designer might also want to select a desirable design in each far-future design and determine which near-future design might be the most suitable baseline (see Figure 6.16d). Such a case is illustrated in Figure 6.25, where a design is selected in each 2025-2035 scenario. The average commonality across the three scenarios is then calculated for each possible baseline in 2015 and the highest scoring design is highlighted by the black dot in BBT 2015. Note that the median, minimum, maximum, or standard deviation of the commonality scores/cost could also be used to find a single baseline.

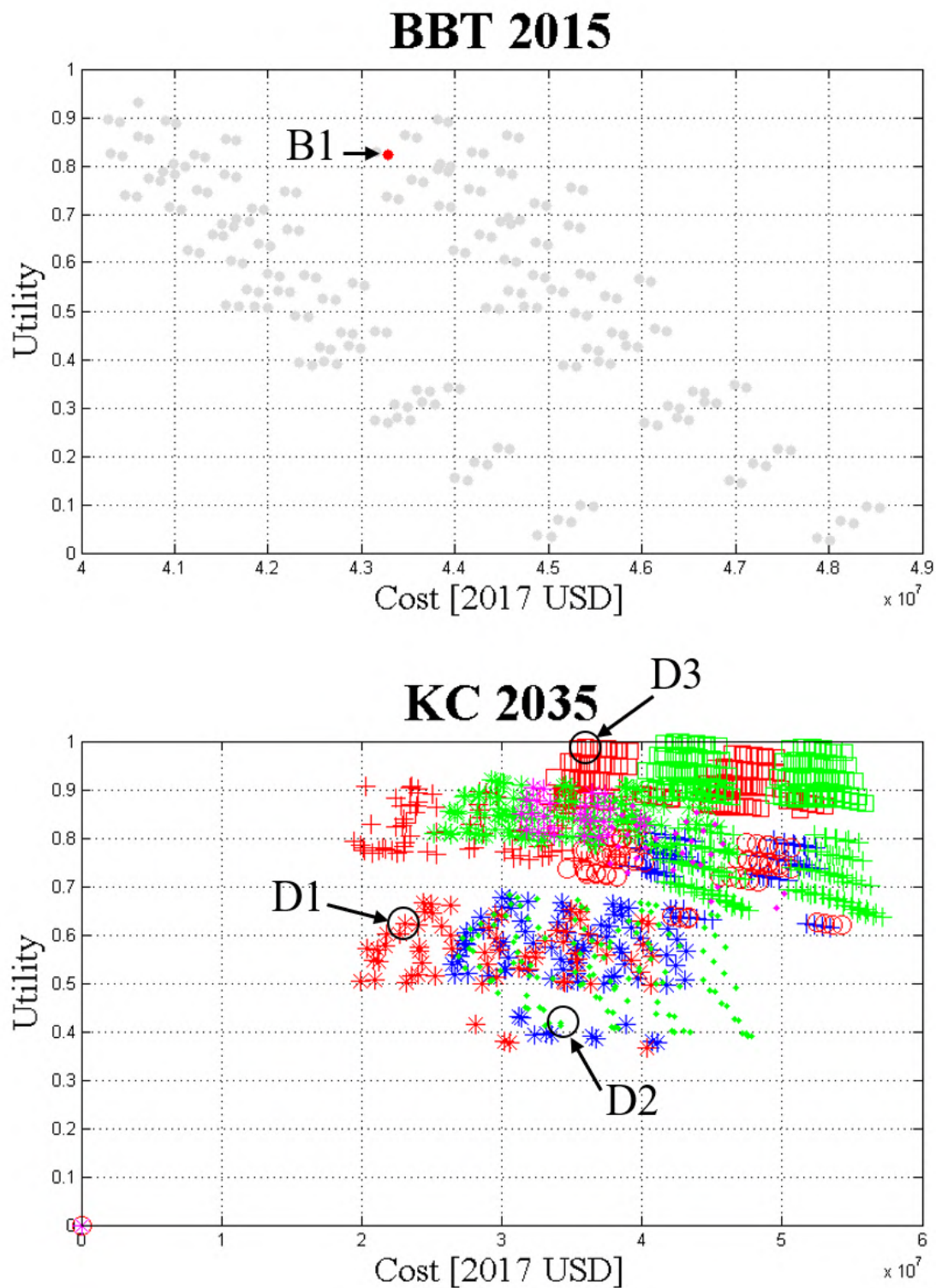


Figure 6.20: Investigating descendant cost in far-future scenario KC 2035, for a selected T-tail baseline (B1) in BBT 2015. Each point represents a single design, whereas the colours and symbols each represent a different configuration (see Figure 6.1 for a guide). Geometry renderings, displaying the predicted commonality between design option B1 and example descendant options D1, D2, and D3, are shown in Figure 6.22.

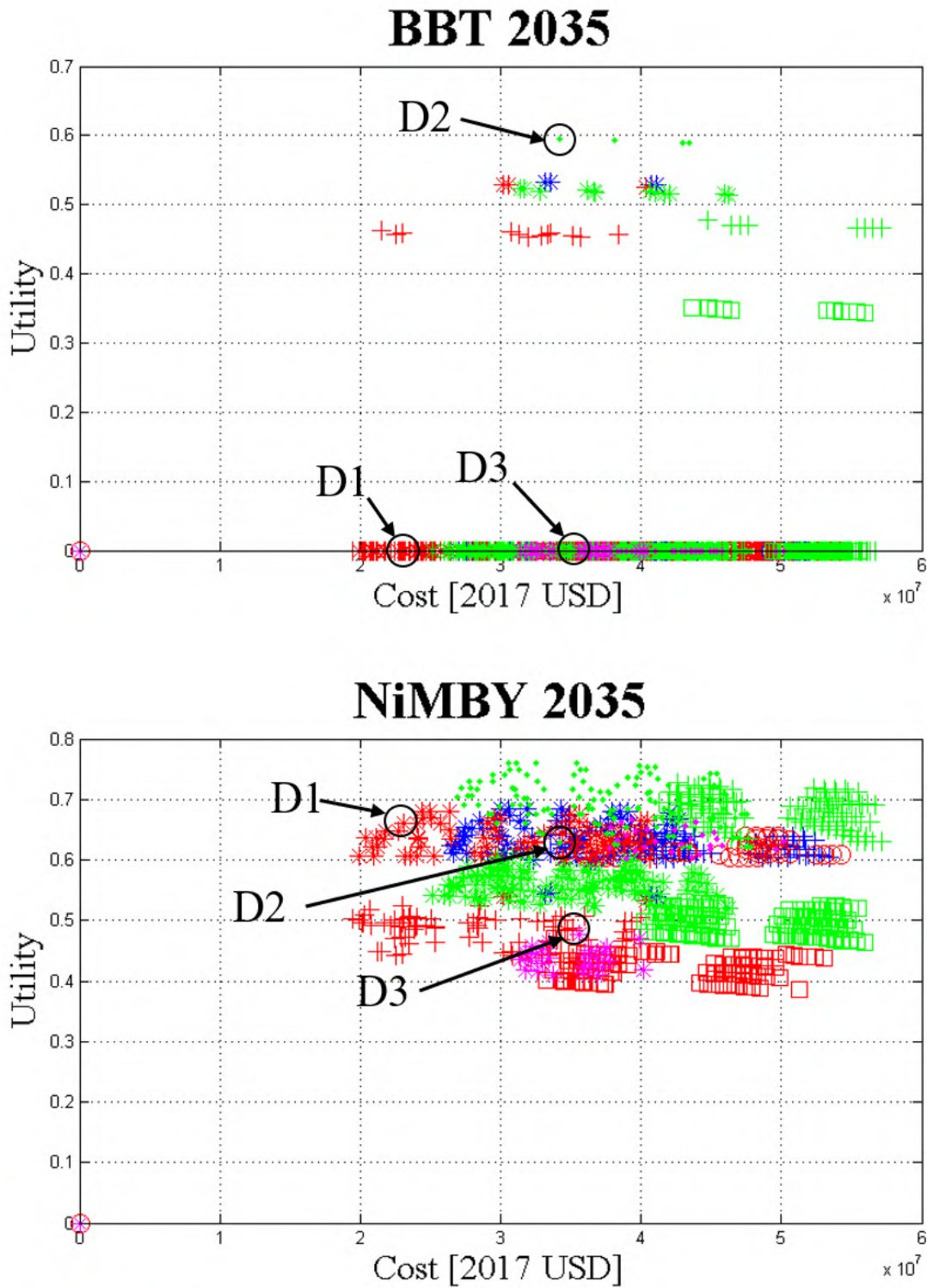


Figure 6.21: Investigating descendant cost in far-future scenarios BBT 2035 and NiMBY 2035, for a selected T-tail baseline in BBT 2015 (B1 in Figure 6.20). Each point represents a single design, whereas the colours and symbols each represent a different configuration (see Figure 6.1 for a guide). Geometry renderings, displaying the predicted commonality between design option B1 and example descendant options D1, D2, and D3, are shown in Figure 6.22.

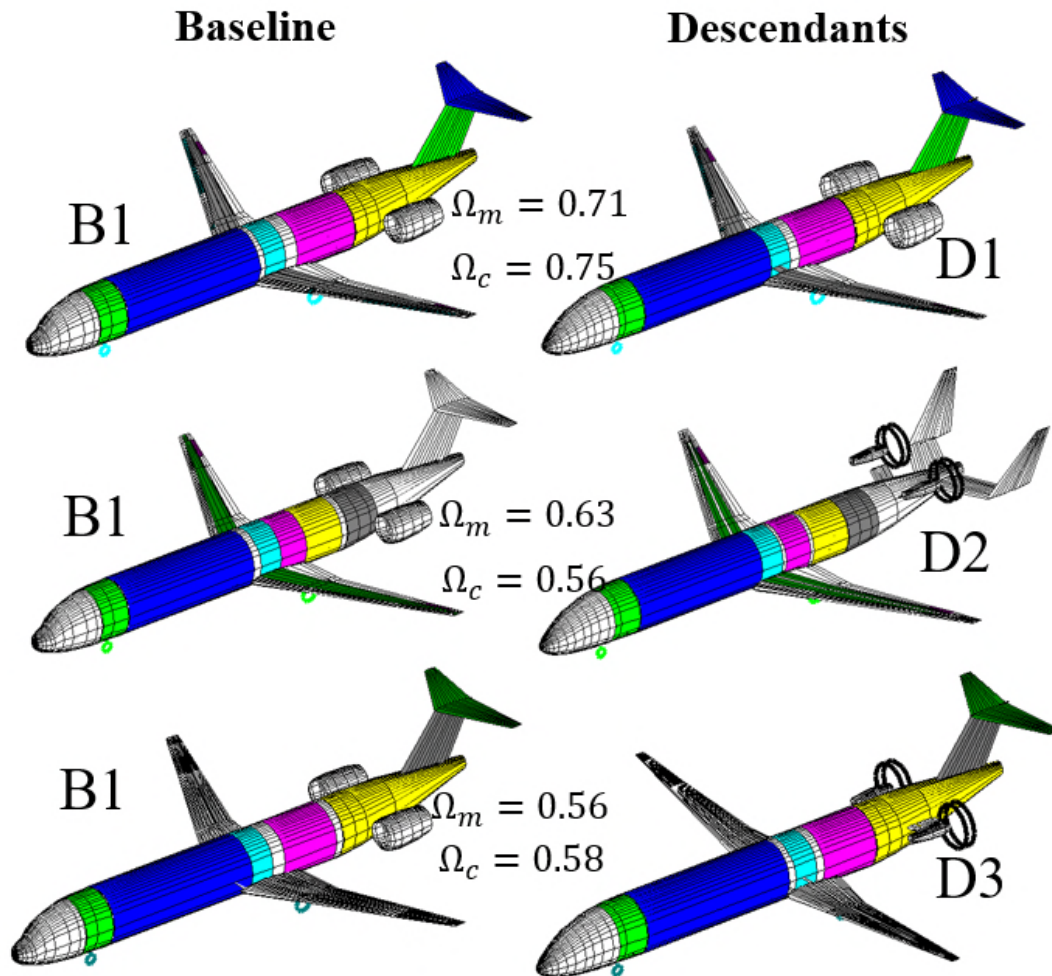


Figure 6.22: Geometry renderings, displaying the predicted commonality between design option B1 and example descendant options D1, D2, and D3 in Figure 6.20 and 6.21.

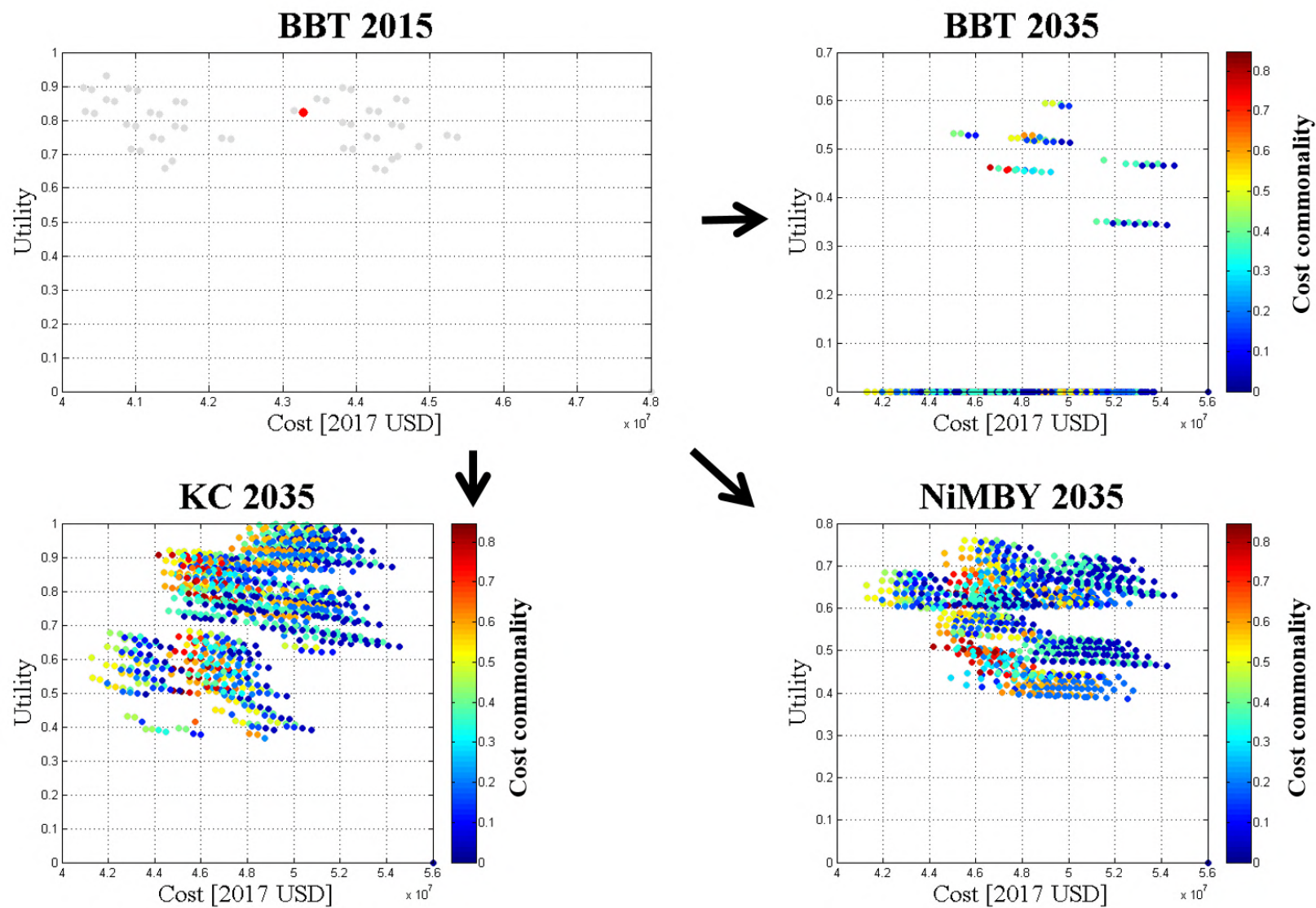


Figure 6.23: Exploring commonality for a selected baseline.

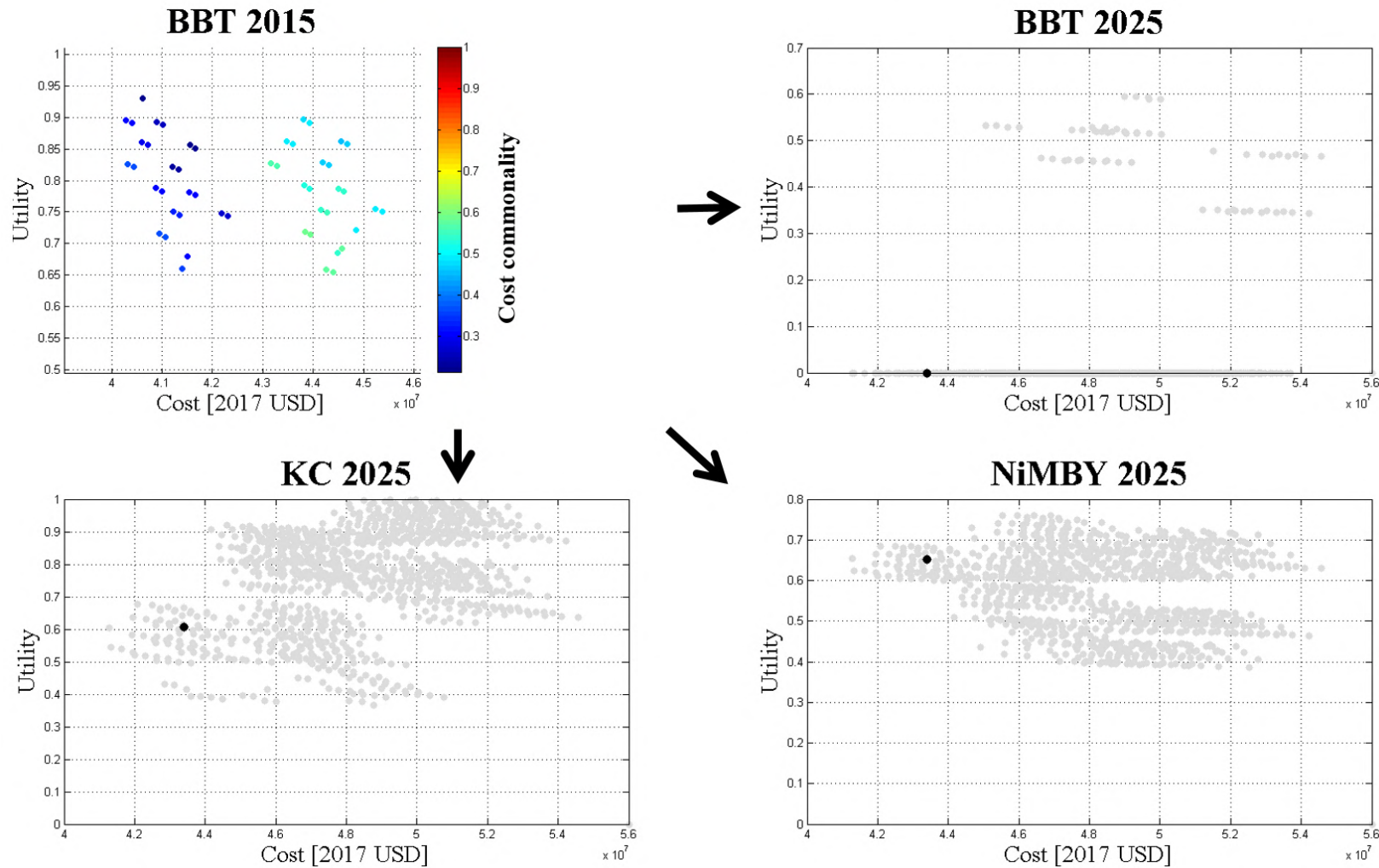


Figure 6.24: ‘Future to past’ exploration for a single descendant.

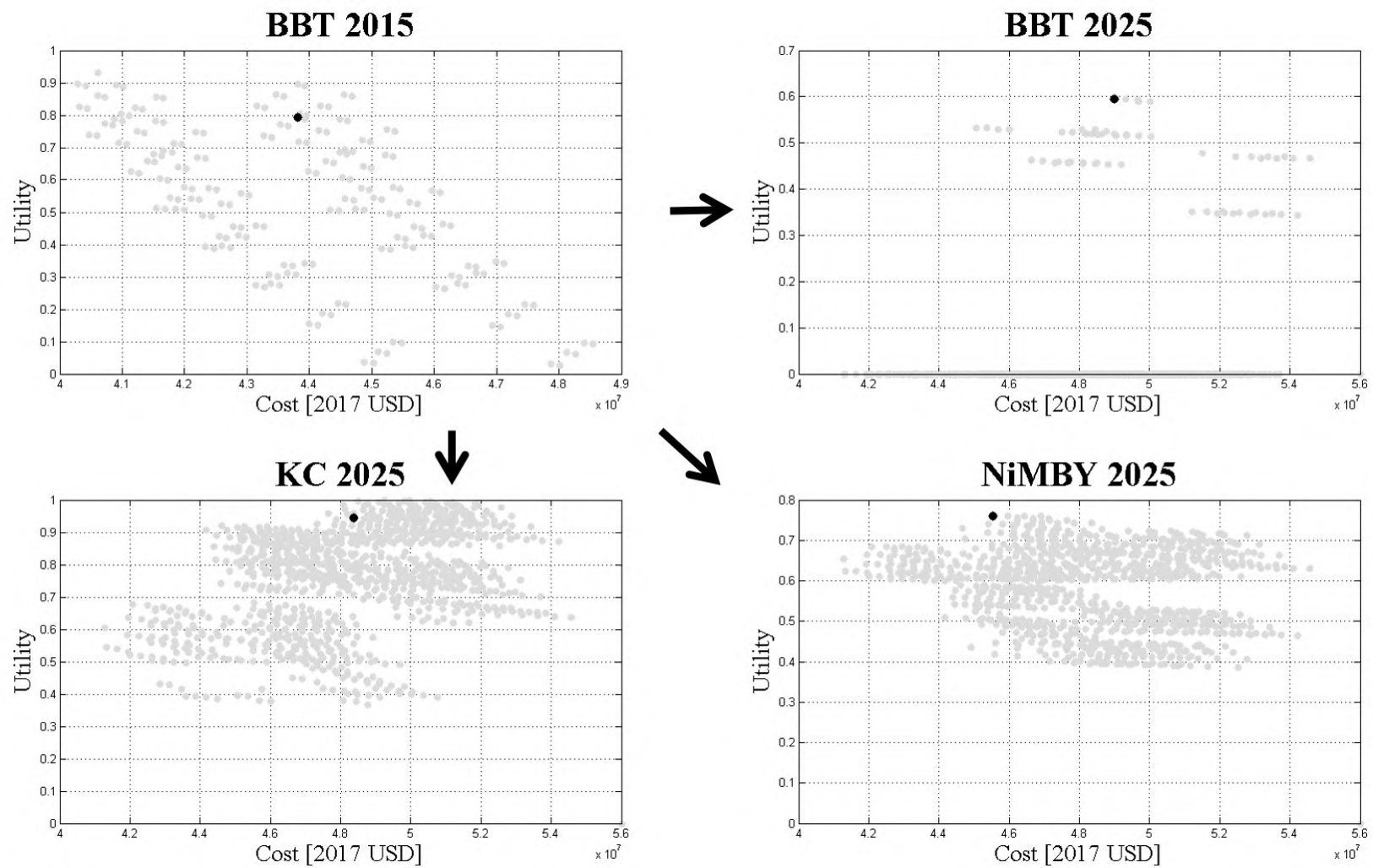


Figure 6.25: ‘Future to past’ exploration for multiple descendants.

6.3.6 Computational Cost

All the commonality prediction (CP) runs were performed on a standard desktop PC with an Intel Core i7-4770 CPU @ 3.40 GHz processor. As already stated, the average computational time for a CP run for the design case study was calculated to be 2.0768 s. This is longer than FLOPS takes to execute a single design point, which, when considering Figure 1.3, means that the use of the commonality prediction module to expedite this particular study is limited. More generally, it implies that the commonality prediction, in its current state of development, is limited to studies in which it is first endeavoured to reduce the size of design space before doing similarity assessment (which the techniques enable), or in which execution time per design point is higher, such as in more detailed, physics-based design studies. The latter are indeed becoming more common, as designers attempt to account for more phenomena earlier in the design process.

However, future work should focus on expediting the CP process. The first avenue would be to investigate discretisation of the airframe elements. It was beyond the scope of this research to fully investigate the effects of discretisation of the fuselage and wing on execution time. For the runs in the studies provided here, the discretisation of the fuselage of about 0.3% of the length of the shorter fuselage and 0.1% of the wingspan of the shorter wing consistently provide expected results in terms of segments selected to be common (as confirmed through visual inspection of the geometry renderings). Studies need to be performed to determine how much this could be increased, without losing accuracy in the commonality predicted, and what the potential benefits in execution time might entail.

6.4 Set-Based Techniques Demonstration

6.4.1 Purpose

The purpose of this section is to provide a demonstration of the evolvability set-based design techniques proposed in Chapter 5. A brief overview of the approach followed is provided below, followed by the results and a brief discussion.

6.4.2 Approach

Each of the techniques presented in Chapter 5 was implemented into a separate MATLAB[®] script. The sizing and commonality results of the demonstration case study from Section 6.2 were then fed to the dominance criteria (Sections 5.3.1 to 5.3.3) in the order shown in Figure 5.1. For the purposes of this demonstration, it was assumed that the input data were deterministic. The cost commonality scores

calculated (see Section 6.3.5) served as the measure of redesign effort, where lower scores reflect higher redesign effort. In other words, redesign effort was set equal to one minus the cost commonality score. Each criterion was applied separately for both the near- and far-future timeframes (except E-admissibility, which is deemed not appropriate for near-future designs). After this, the ‘constrain and union’ strategy (Section 5.4) was applied to the remaining designs to further prune the evolvability design space.

6.4.3 Results and Discussion

For the demonstration case study, no designs were removed as a result of applying the interval dominance criterion. This means that none of the designs in the study dominated another design in terms of utility, cost, and redesign effort to all/from all far-future/near-future design options. The reason for this is that each design has some designs in either the 2015 or 2025-2035 timeframes which have zero commonality with it and none therefore dominate over the whole redesign effort interval.

Applying maximality (see Section 5.3.2) using the less strict form of the criterion (i.e. with only inequalities) also did not result in any pruning. However, once two of the three inequalities were allowed to be equalities, a single design could be removed from the 50 potential baseline options and 475 designs could be removed from the 1,444 designs in the far-future timeframe. These are depicted by the greyed-out design points shown in Figure 6.26. The execution time was 0.648 s for filtering the near-future designs and 8.88 s for the far-future designs (on a standard desktop computer with an Intel Core i7-4770 CPU @ 3.40 GHz processor).

The next criterion was E-admissibility (Section 5.3.3), which was applied to the pruned set of far-future options (i.e. without the 475 previously filtered-out designs). The result is shown in Figure 6.27. A further 624 designs could be removed using this criterion. The execution time in this case was 369.2 s, which, is considerably longer than with maximality.

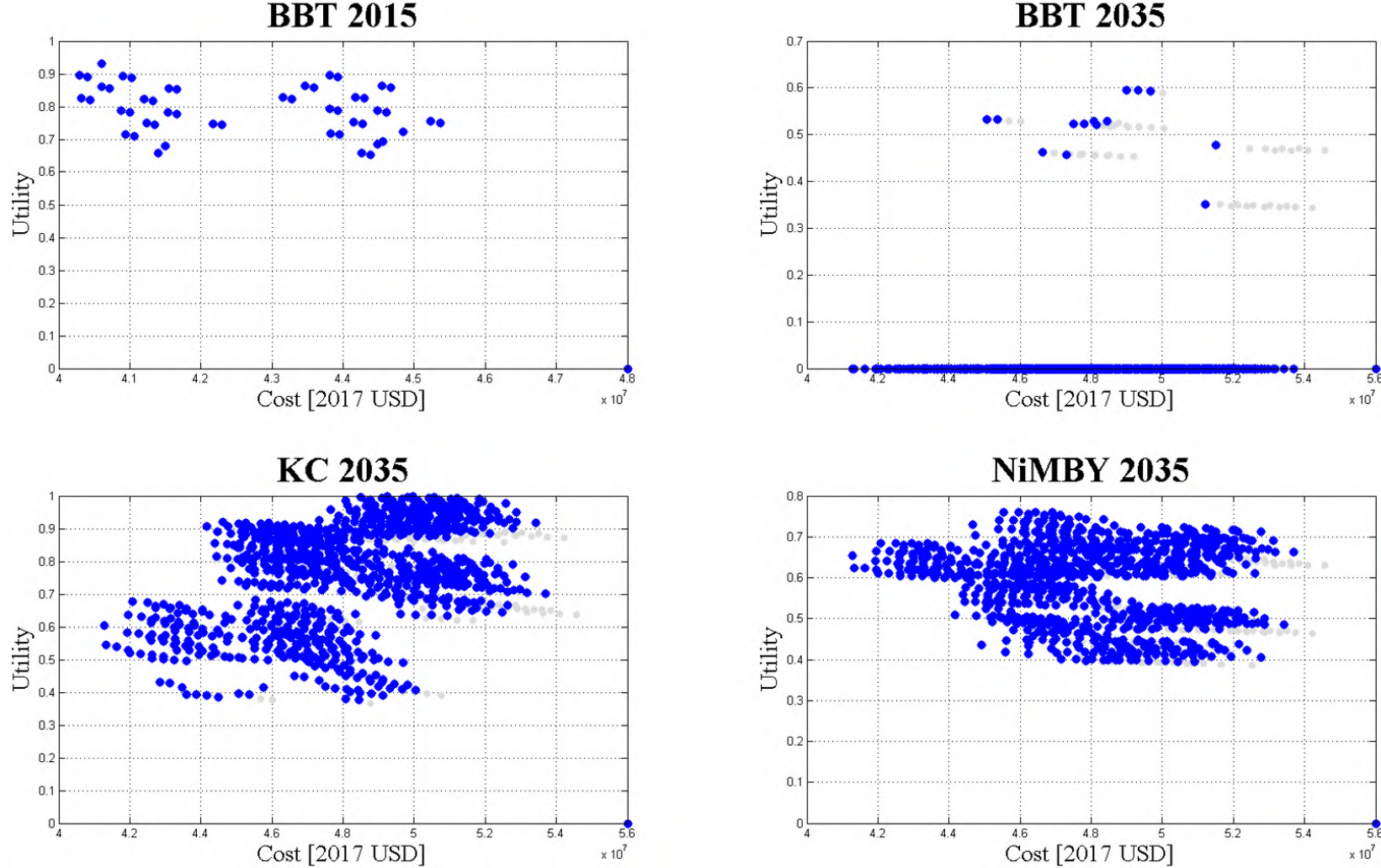


Figure 6.26: Applying the maximality criterion to far-future designs.

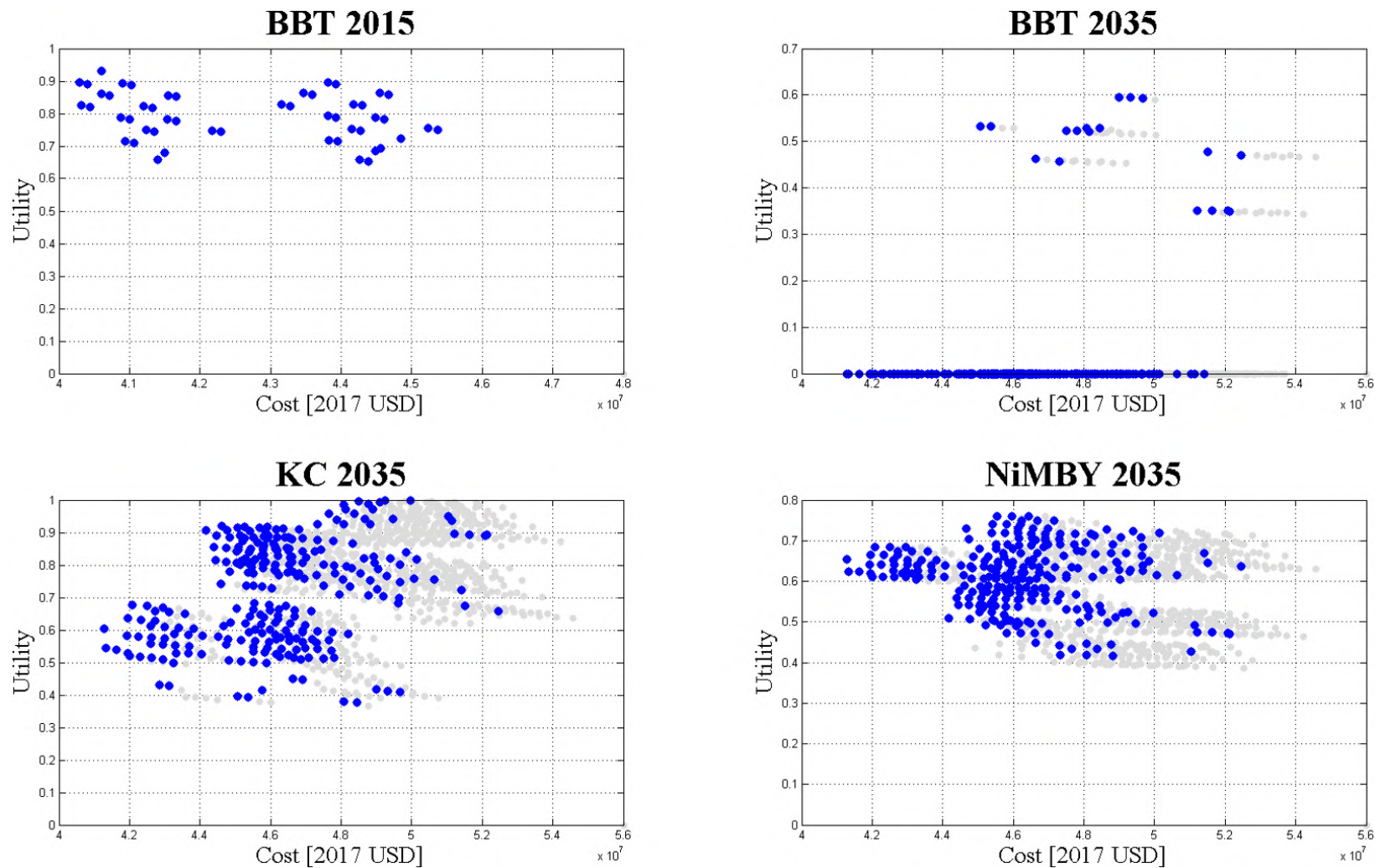


Figure 6.27: Applying E-admissibility to the far-future designs.

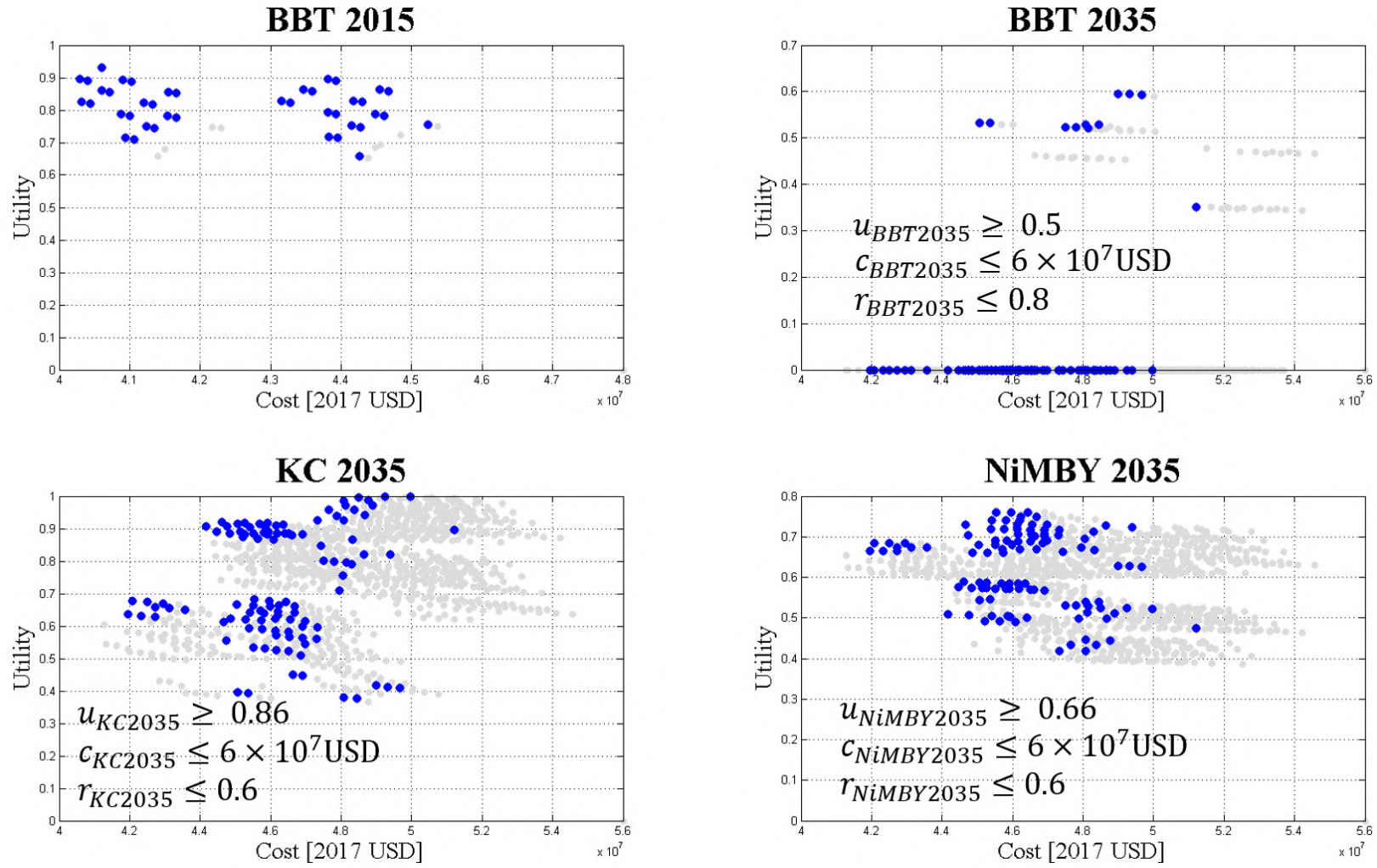


Figure 6.28: Applying the constrain and union strategy to the evolvability design space.

The results of applying the constrain and union strategy (see Section 5.4) are shown in Figure 6.28. In this example case, the Fuzzy-Pareto number was set such that 20 designs remained in the near-future timeframe for both the C1 and C3 configuration-technology combinations (i.e. the number of designs in the BBT 2015 have reduced to 40). The number of far-future design options has reduced to 178 as a result of specifying the minimum utility, maximum cost, and maximum redesign effort values for each scenario as shown in the figure. These can be varied to further increase/reduce the number of designs to be filtered out. Note that some designs remain in a particular scenario despite not meeting the requirements of that scenario. This is simply because they meet the requirements of one or more of the other scenarios and were therefore not discarded. The execution time in this case was 0.577 s, which makes it suitable for interactive exploration.

The above selection of results demonstrated how the set-based design techniques could be applied to a conceptual evolvability study. As could be seen, by applying the techniques in a consecutive manner, the design space could systematically (and automatically) be pruned of less promising design options. The constrain and union strategy is especially amenable to interactive exploration and varying the minimum/maximum values for utilities, costs and redesign effort would enable the designer to rapidly investigate how the evolvability design space would be affected by changes in these factors.

6.5 Industry Feedback

6.5.1 Purpose

The final part of the evaluation involved a limited initial application and success study. This was performed by obtaining feedback regarding the usefulness and industrial relevance of the proposed techniques from aircraft conceptual design and modelling and simulation specialists from Airbus UK. The objectives were specifically to obtain feedback from this group regarding:

- the validity of the assumptions;
- the validity of the techniques themselves;
- the usefulness/relevance of the approach; and
- how the techniques could be improved.

6.5.2 Approach

A feedback questionnaire was created that combined closed-form Likert-type questions (questions that probe for the extent to which a respondent agrees with a

statement) with more open-ended questions. A copy of this questionnaire can be viewed in Appendix D.

Question 1 constituted the Likert questions. Question 1.1 was posed such that one of the basic assumptions of the research (i.e. that similarity could be used to provide a reasonable indication of redesign effort) could be validated. Questions 1.2 and 1.3 were set to provide an additional confirmation of the validity of the results presented earlier in this chapter. Questions 1.4 to 1.8 were formulated to establish whether the techniques (if implemented into a robust software application) would add practical value to the individual user and the company as a whole. Question 2 was set such that the respondents could highlight specific aspects of the techniques that they liked or disliked. Finally Question 3 was set to obtain any further comments or suggestions, which could aid in future research.

To perform the evaluation, the author visited the Airbus UK site at Filton on the 30th of January 2018. A two-hour session was held with a group of four specialists. The session was structured as follows:

- Introductory presentation (25 min). This was to introduce the respondents to the work and demonstrate to them the results presented in this chapter.
- Questions and discussion (30 min). Here, the respondents were given some time to ask clarifying questions and to freely discuss the work.
- Questionnaire (15 min).
- Open questions/further discussion (50 min). During this stage, further questions, especially regarding the soundness of the assumptions made, were asked of the respondents.

The credentials of the four respondents are summarised in Table 6.11.

Table 6.11: Respondent information.

	Job Title	Years of relevant experience
Respondent 1	Expert for Modelling & Simulation methods and tools	20+
Respondent 2	Future Projects Engineer	18
Respondent 3	Future Projects Engineer	9
Respondent 4	Modelling and Simulation Specialist	30

6.5.3 Results and Discussion

The responses to the Likert questions are summarised in Figure 6.29. As can be seen, the respondents agreed or strongly agreed with all of the statements, except for statements 1.4 and 1.8. For statement 1.4, respondent 2, who chose ‘neither agree or disagree’, indicated that none of their studies were performed in this way yet and therefore that it would be difficult to predict. For 1.8, the same respondent said that he is not sure yet, but that the methods would probably be essential for some of their approaches.

The responses to Question 2 were all positive. Respondent 1 said that the commonality prediction techniques form an “interesting approach”. Respondent 2 wrote: “I like the fact that this seems to allow comparison between different a/c [aircraft] on a neutral basis” and “It’s possible that, for some of the things we’d like to do, this could be a very useful way to rigorously compare a/c at early stages.” Respondent 3 was interested in the follow-on questions for the “next PhDs” and indicated that the “Concepts [were] explained at the right level for the audience”. Finally, Respondent 4 added that the approach “Provides a “simple method” to explore a complex topic” and that it “Should be relatively easy to integrate with other tools”.

The responses to Question 3 were especially useful for obtaining ideas for further improvement. Respondent 1 suggested the “Use of Structured Expert Judgement to apply probabilities to future scenarios – [and] also [for] the determination of the parameters in your distance measure”². Respondent three provided several suggestions, such as providing a means to ensure that the “subjective ‘penalty’ or ‘weighting’ functions” remain “transparent and adaptable for end users” and that this could be “couple[d] with expert elicitation processes”. Respondent 4, indicated that “It would be interesting to see it applied to another area (e.g. industrial costs) to show the generic nature of the method” and that “It would also be interesting to see how it could be used to generate/help design product families”. As can be observed from these responses, the suggestions generally relate to i) making use of expert elicitation methods to obtain the required information for the techniques and ii) making the methods more generic and adding additional features and functionality. Finally, the respondents agreed that the important assumptions made during the research were reasonable for the stage of design considered.

To conclude, the respondents agreed that the techniques would likely be useful, allow them to perform more efficient and comprehensive evolvability studies and

²The distance measure is a technique which may expedite similarity detection and which the author is currently investigating.

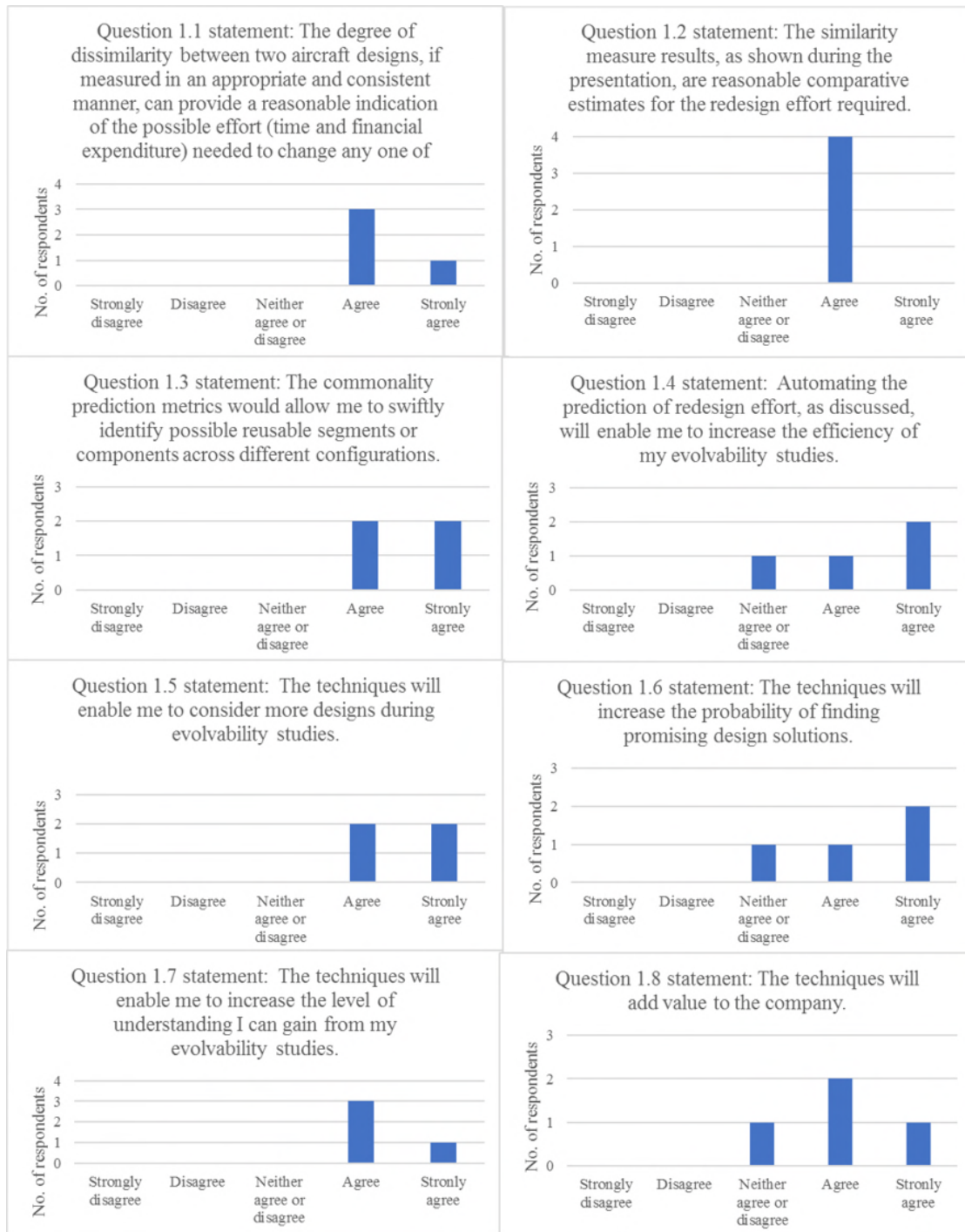


Figure 6.29: Responses to Question 1: Please indicate to what extent you agree or disagree with the following statements, by marking the appropriate box.

that these would add value to Airbus. This initial application and success evaluation was therefore successful. However, further evaluation would need to be performed, which should involve the intended users actually using the software implementation. This would undoubtedly lead to further refinement and is the subject of future work.

6.6 Conclusions

This chapter covered the evaluation of the research. In line with the Design Research Methodology presented by Blessing and Chakrabarti [18], this evaluation consisted of three stages: the support evaluation, application evaluation, and success.

The results from the support study for the airframe similarity assessment framework indicate that the commonality prediction techniques (CPTs) function correctly and provide expected results. In particular, it was observed that the CPTs:

- correctly detected full commonality when given the same aircraft as input both as the baseline, as well as the descendant. This was as a ‘sanity check’.
- provided commonality assignment that matched publicly available descriptions of several current and historic aircraft of both Boeing and Airbus.
- provided expected assignment of commonality across aircraft with dissimilar configurations.

The CPTs therefore fulfil the need for which Objective 1 of this research was formulated, i.e. to develop techniques that could enable automatic similarity assessment across different airframes.

However, it was determined that, for the commonality prediction runs on the design case study, the average execution time (for a single run) on a standard PC (Intel Core i7-4770 CPU @ 3.40 GHz) was about 2.1 s. This is longer than it takes to execute FLOPS for a single design point and implies that the usefulness of the commonality prediction is currently limited in cases where design execution time is short. However, the techniques could still substantially decrease overall time in cases where i) it is first endeavoured to reduce the size of design space before doing similarity assessment (as they enable this type of approach), or ii) where execution time per design point is longer, such in more detailed, physics-based design studies (which are becoming more common).

For the set-based design techniques, it was demonstrated how the design space could be progressively pruned using the proposed criteria. Interval dominance provided no filtering as none of the designs dominated another in terms of intervals

for cost, utility, and redesign effort. Maximality and E-admissibility successfully eliminated several inferior alternatives. E-admissibility took substantially longer but, as expected, removed more designs. The constrain and union provided an effective additional strategy and, because of its short execution time (less than 0.6 s), would be useful in interactive exploration. The second objective of the research (to devise/adapt set-based design techniques for use in evolvability exploration) was therefore achieved as well.

The application and success evaluations were performed by obtaining feedback from a panel of practising aircraft conceptual designers from Airbus UK. Although this only formed an initial application and success evaluation, the feedback from the panel was positive. In particular, it was indicated that the proposed techniques would likely enable new ways of exploring evolvability, and provide for more efficient and comprehensive evolvability exploration during conceptual design.

CHAPTER 7

Summary and Conclusions

7.1 Introduction

In this chapter, the main body of this thesis is concluded. First, a summary of the research is provided in Section 7.2, specifically illustrating how the aims and objectives were achieved by means of the techniques developed. This is followed in Section 7.3 by an in-depth discussion of the research findings. In Section 7.4, the contributions to knowledge and engineering practice are highlighted, whereas suggestions for future work are provided in Section 7.5.

7.2 Research Summary

The aim of the research presented in this thesis was to develop computational techniques that would enable civil transport aircraft designers to explore the evolvability of their airframe designs more efficiently and effectively during the conceptual design stage.

The research followed the four stages of engineering design research identified by Blessing and Chakrabarti [18]. The first stage (Research Clarification – RC) consisted of a literature review to identify the main industrial needs pertaining to evolvability exploration at conceptual design stage in order to formulate an aim and objectives that are academically and practically worthwhile, as well as realistic. The aim was therefore formulated based on needs related to the shortcomings of existing techniques, which were identified during the RC stage. These

needs were translated into Research Questions and, subsequently, into objectives. In the subsections that follow, these needs, Research Questions, and objectives are restated and it is summarised how the proposed techniques developed met/answered/achieved them.

The second stage (Descriptive Study I –DSI) involved further study of the literature and case studies to further improve understanding of the phenomenon of evolvability exploration in aircraft conceptual design. The purpose was to understand how evolvability is achieved on actual aircraft, to identify the existing state of the art in evolvability exploration techniques, as well as to identify possible tools and methods from other fields that could be employed to achieve the aim and objectives.

The first part of DSI was the literature review (which was also part of the RC stage), which indicated that several methods for evolvability exploration existed. However, it was shown that these could be improved upon with new approaches that could decrease the computational cost involved, could automatically suggest possible design change rules rather than the user having to pre-specify them, and enable a more systematic approach to filtering the evolvability design space. In addition, the literature review indicated that knowledge-based engineering, a longest common subsequence approach, and set-based design criteria could aid in developing techniques to achieve the aim, as will be described in the subsections below.

The second part of DSI involved case studies (Chapter 3), which were performed on several existing aircraft to determine which design changes are actually possible and reasonable in the context of evolvability. The aircraft studied included several members from aircraft families produced by Airbus, Boeing, and McDonnell-Douglas. The study resulted in a number of interesting observations, as well as a summary of design changes achieved for actual aircraft in the main airframe design parameters, such as fuselage length and diameter, wingspan, sweep, and chord, undercarriage layout and length, and empennage surface area, amongst others.

The next stage of the research (the Prescriptive Study) involved devising the computational techniques to achieve the aims and objectives, whereas the final step (Descriptive Study II – DSII) was concerned with evaluating the consistency, accuracy, and industrial usefulness of these techniques. A description follows of how the aim and objectives were achieved by these techniques. This will, in turn, be followed by a discussion on the evaluation of the research (i.e. DSII).

7.2.1 Objective 1

The first need identified from the RC stage and DSI was to reduce the computational cost associated with current techniques for evolvability exploration during conceptual design. It was demonstrated that, if the baseline and far-future design spaces were to be sampled separately, significant cost savings could be obtained. However, this would require a means to estimate similarity between airframes, which could be used as a prediction for redesign effort. Similarity across two airframes, for the purposes of this research, is determined by the degree of commonality between the airframes.

Related to this, another need (or more specifically, an opportunity) was identified. Currently, evolvability exploration is usually performed by pre-specifying change rules to a selected baseline. This could become cumbersome if there were many different configurations to be studied. It would be advantageous if the designer could obtain suggestions for change rules from the exploration study, rather than pre-specifying them. This would enable new types of evolvability exploration activities, such as selecting a far-future option with a novel configuration and subsequently determining how baseline options could evolve to it. To obtain such suggestions for change rules would, again, require some means of automatic similarity assessment between designs, especially if this similarity assessment could indicate which components or airframe segments could be made common.

The first research question was therefore formulated as follows:

Research Question 1: How can similarity between airframes be assessed automatically to predict redesign effort?

This was reformulated into the following objective:

Objective 1: To devise computational techniques that could assess similarity between airframes automatically.

Based on the information obtained from DSI, a computational framework for automatic airframe similarity assessment was developed to meet Objective 1. This framework was detailed in Chapter 4. The framework can be considered an architecture for a knowledge-based engineering (KBE) application, although it was not formally developed as such.

As with any KBE application, the framework comprises two aspects, namely knowledge representation and a set of inference mechanisms. The knowledge representation aspect of the framework provides a structured means to store information

on the airframes to be studied. A generic aircraft object model and a template for aircraft design structure matrices (DSMs) were introduced for this purpose. The object model stores salient geometrical, material, and process attributes of the aircraft, whereas the DSM dictates how the different constituent components are attached to each other.

The inference mechanisms of the framework (referred to as modules) process this knowledge for pairs of airframes to determine their similarity to each other. The inference mechanisms proposed include an ‘airframe partitioning module and an ‘airframe commonality predictor’. The partitioning module determines which pairs of airframe components from two selected parent aircraft should be compared, based on the type of components and how they are connected to the other components in the airframe. These pairs are then dispatched to the commonality predictor, which is discussed below.

It is vital to note that, when using the similarity assessment techniques proposed here to enable separate sampling of the baseline and far-future design spaces, the performance implications of having commonality with a baseline will not be captured. These were shown to be small (see Chapter 1.2.1), but not negligible. It was outside the scope of the research to address these implications fully, however suggestions are provided in the future work discussion (provided in Section 7.5) on how to deal with them.

7.2.1.1 Airframe Commonality Predictor

The commonality predictor estimates the degree of mass or cost commonality between two given airframes. It provides as output an overall estimate of commonality, which is calculated by dividing the mass/cost of the common components/segments in these two airframes by the total mass/cost. Whether cost commonality can be calculated depends on whether an appropriate cost model is available, in which case the output could also include the predicted cost-reduction benefits of the commonality, if implemented.

For airframe ‘body’ components (fuselages and nacelles), the commonality prediction determines whether pairs of longitudinal (x -direction) segments (from the parent body airframe pair being studied) could be identified that are similar enough to be considered common. This is achieved by an algorithm that was proposed to solve a longest common subsequence problem posed for airframe body components.

For wing commonality, a multi-objective optimisation procedure was proposed to determine the span for a wingtip-extension and/or root-insert that would minimise i) the length of the root-insert; ii) the difference in chord length across the span;

and iii) the total difference in wing planform area. If the chord extensions are too large, the wings cannot be made common. After determining the span of the tip-extension/root-insert, the carry-through structure, wing-box, fixed leading- and trailing-edges, ailerons, and high-lift devices are discretised and compared spanwise to calculate the commonality for each discretised spanwise segment. These are then summed to determine the overall wing commonality.

Finally, for empennage surfaces, the planform areas are compared. If the possible descendant has a larger area than the baseline, this larger area is achieved by adding a root-insert and/or tip-extension to the baseline surface. If the area of the baseline is larger than that of the descendant, the descendant adopts the baseline area. For undercarriage, the length difference of the main gear is calculated to determine if the undercarriage could be made common across the two airframes, as well as what the additional effort of increasing the length could entail. Commonality for miscellaneous components, such as struts, booms, and pylons are only assigned if the difference in salient design parameters describing them are within user-specified tolerances.

The common mass/cost for all the components and segments of the airframes are then summed and divided by the total mass/cost to obtain a commonality score. This score is only meaningful when estimating the redesign effort of different descendants from the same baseline, or from different baselines to the same descendant. Therefore, the commonality must be weighted in some manner so that all baseline-descendant pairs in a study could be compared with each other to determine the relative redesign effort. To do this, the common mass/cost for each pair could be divided by the total maximum mass/cost of all the pairs in the study, or the monetary value of the potential cost savings because of commonality could be calculated.

The development of the airframe similarity assessment framework, along with the airframe partitioning module and knowledge representation, answered Research Question 1. Specifically, the similarity between airframes can be assessed automatically by the combination of airframe partitioning and component commonality prediction algorithms, given the salient geometry, material, and configuration information that could be obtained from the aircraft object models and DSMs. Objective 1 is therefore achieved, and enables a possibly significant reduction in computational cost, because the baseline and far-future design space can now be sampled separately, and the redesign effort estimated afterwards. More baseline and far-future designs could therefore be studied with a given amount of time and computing resources, which increases the likelihood of finding promising evolvable designs. The savings in computational cost does, however, depend strongly on the

ratio of commonality assessment time and design point execution time. Even if this is high, significant saving could still be obtained, as the design space can first be pruned based on other criteria. The development of the commonality prediction techniques therefore contributes to achieving the aim.

Furthermore, as the commonality prediction techniques identifies which airframe components/segments could be made common, it essentially provides the designer with suggestions as to how the baseline could be redesigned. The designer therefore does not have to pre-specify change rules. The provision of this enabler meets the need of obtaining suggestions from the evolvability study on which change rules could be applied to achieve certain far-future designs from a selected baseline, or vice versa.

7.2.2 Objective 2

Another need identified from the RC and DSI stages was that there appeared to be a lack of suitable filtering techniques to eliminate inferior design options under consideration systematically during evolvability studies. The literature survey conducted during these stages indicated that a set-based design approach could be promising in providing such techniques. The second research question was therefore stated as follows:

Research Question 2: How can set-based design methods be devised and/or adapted for evolvability exploration studies?

And the second objective was formulated from this as follows:

Objective 2: To devise new and/or adapt existing set-based design criteria to enable systematic filtering of the evolvability design space.

To achieve this objective, the existing set-based design criteria of ‘interval dominance’, ‘maximality’, and ‘E-admissibility’ were extended to be used in evolvability design studies (see Chapter 5). Interval dominance was extended for evolvability to determine if there were design options, or subsets of design options, that had lower maximum utility, higher minimum cost, and higher minimum redesign effort (to either far-future options, or from baseline options) than the minimum utility, maximum cost, and maximum redesign effort of at least one other design/design set. If such an option/subset could be found, it could be eliminated.

Maximality is a stricter criterion than interval dominance and could therefore potentially find more designs to discard. However, it is also more computationally expensive. Maximality was extended for evolvability to determine whether a

design/subset was dominated in utility, cost, and redesign effort to all descendant-s/from all baselines in all redesign cases by another design/set. If true, the former design/set could be removed from the study.

In E-admissibility, it is determined whether, for every redesign case, a selected design/set is dominated by at least one other design/set in terms of utility, cost, and redesign effort. If indeed, it could be removed from the study. E-admissibility is at least as computationally expensive as maximality, but will potentially remove more designs. It was demonstrated that it is not appropriate to apply E-admissibility for eliminating baseline designs, as the decision of which far-future option to evolve to would not necessarily be made before deciding on which baseline to pursue.

In addition to extending these set-based criteria, a strategy for further reduction of the evolvability design space was introduced. This involves constraining the near- and far-future design space in terms of utility and cost, as well as the maximum redesign effort for each scenario. The union of the sets of far-future design that adhere to these constraints are then retained for further investigation.

Collectively, these techniques provide a powerful means to swiftly, but systematically reduce the evolvability design space. This will allow designers to focus their resources on more promising designs, while not prematurely converging to a single design option. Objective 2 is therefore met with these techniques, and Research Question 2 is answered. These techniques thereby contribute to the aim, as they enable more designs to be down-selected within a given time, increasing the efficiency of the evolvability exploration process.

7.2.3 Evaluation

For this research, only an initial evaluation was possible within the given timeframe. This evaluation consisted of three stages, consistent with the general engineering design research methodology, as introduced by Blessing and Chakrabarti [18]. These stages are the ‘support’, ‘application’, and ‘success’ evaluation steps.

The support evaluation involved testing (verification) of the techniques to ensure they were consistent and produced accurate results. To do this, all the techniques developed were implemented into MATLAB[®] scripts. A specific application case study – a design study for a narrow-body environmentally friendly airliner – was also developed.

The airframe similarity assessment framework was then applied to recreate several existing examples of evolvability in civil transport aircraft to determine whether they could detect accurately which airframe components/segments were kept common in these examples. The examples included members from the Boeing 707,

727, 737, and the Airbus A320, A330, and A340 families. The results of this study indicated that the commonality predicted by the commonality prediction module matched publicly available descriptions of the commonality within these aircraft families well. The commonality prediction module was subsequently applied to pairs of conventional and novel configurations to demonstrate that it could predict commonality across radically dissimilar configurations. Finally, the techniques were applied for the design case study to show how the results of the similarity detection could be exploited during evolvability exploration. Because no publicly available information exists for the unconventional configurations, the results were demonstrated to a panel of conceptual aircraft design specialist from Airbus UK. The panel indicated that the methods provided expected results and that the assumptions made during the development of the research were reasonably sound for conceptual design.

These studies also indicated that the computational cost of estimating commonality was small, but needs to be reduced to ensure usefulness. For a commonality prediction run, the time on a standard PC (Intel Core i7-4770 CPU @ 3.40 GHz) was about 2.1 s. This is longer than it takes to execute FLOPS for a single design point, which means that the usefulness of the commonality prediction is currently limited to cases where i) it is first endeavoured to reduce the size of design space before doing similarity assessment, or ii) where execution time per design point is longer, such in more detailed, physics-based design studies (which are becoming more common).

Finally, algorithms embodying the set-based design techniques developed during the research were applied to the design case study to determine whether they functioned correctly and to demonstrate their usefulness. The results of these test cases demonstrated that they did indeed function correctly and it was shown how large numbers of designs could be filtered from the evolvability space.

The purpose of the ‘application study’ was to determine, if the techniques were to be adopted (i.e. applied), whether they would have the intended effect of increasing efficiency and effectiveness (i.e. achieve the aim of the research). It was shown already that the objectives were met through the techniques developed and that the aim was therefore achieved. However, it is still desired to know what the result would be if the techniques were actually implemented in an industrial setting. This could not be investigated directly, as such a study could take years and involve risk on the part of the aircraft manufacturer. The panel of design specialists from Airbus UK was therefore consulted again to evaluate the potential implications of applying the techniques to their work. This panel concluded that the techniques would likely enable them to explore more designs within a given time and increase

the probability of finding promising designs. They stated that it provided them with a structured approach to explore evolvability and that it should be easy to integrate the developed techniques with the existing tools being used in the company.

Finally, the success evaluation was performed to determine if the results of the research would add value if implemented. To do this, the design specialist panel was consulted again. The panel agreed that the research could add value, as it would provide a rigorous approach to study the evolvability of aircraft of different configurations. They also indicated that it enabled new approaches to studying evolvability.

Although an initial evaluation, the results proved promising. Future work will have to focus on testing more configurations and a more rigorous comparison with existing data. An especially important aspect of future work is to address the issue of the performance error that could be obtained when sampling the far-future design space as clean-sheet designs. These are discussed in the Future Work section, provided in Section 7.5.

7.3 Research Findings

In this section, the findings of the research are summarised. These include results that contributed directly to the aim and objectives, as well as other findings that are of interest. The findings are as follows:

1. As was demonstrated in Chapter 1.2, it was found that, if the baseline and far-future design spaces were to be sampled separately for an evolvability exploration study, significant savings in terms of computational cost could be obtained.
2. However, this could cause inaccurate representation of the merit of the descendant aircraft. This is because the effect on performance of having commonality with baseline designs would not be accounted for in such a case. This problem was investigated further and, in Chapter 1.2, it was demonstrated that an aircraft having full airframe commonality with a legacy design, but with new propulsion, was found to have weights, mission fuel, and field-length requirements within 2% of that of a clean-sheet design with the same propulsion and geometrical dimensions. Even though this is an extreme case, these results are still within the range of conceptual design variation. However, they are certainly not trivial, and the user of the proposed techniques should be cognisant of this issue when interpreting results. Techniques are suggested in the future work section to address this issue.

3. A case study performed on existing aircraft evolution examples (Chapter 3) resulted in the findings that follow. These findings were employed in the development of the methods presented in Chapter 4.
 - (a) Changes to wing geometry seem to normally be limited to only an addition of wingtip span extensions, or the modifications to high-lift devices. However, in several cases, span-extending root-inserts and chord-extensions (and even a reduction in chord), were incorporated.
 - (b) Modifications to empennage surfaces are frequent and usually involve tip-extensions, root inserts, or scaling of the whole surface.
 - (c) Changes to structural strength were common. This is normally performed by changing the material gauge. Generally, however, the overall structural layout seems to remain unchanged across generations and even ‘new’ designs.
 - (d) In the early jet-age (1950s and 60s), it seemed that the approach to derive derivative designs was somewhat ad hoc and large design changes were frequent.
 - (e) Later, two distinguishable patterns started to appear – a ‘leap-and-branch’, where a ‘leap’ (a new aircraft, or a major derivative) is followed by a ‘branch’ upwards (a stretch at the expense of range) and a branch-down (a shrink, gaining range). The second pattern is a ‘Z’ shape (also discussed by other authors), where the range of the baseline is systematically increased, followed by a stretch where the range gained is traded for payload. The patterns appear because the OEMs try to meet diverse customer needs, accommodate new technology, and remain competitive.
 - (f) Mimicry (using similar design practices) was found to be pervasive across the three manufacturers. The mimicry between the A300 and A310 and between the Boeing 787, 737 MAX, 747-8, and 777X aircraft are particularly notable examples.
 - (g) The changes made to baselines were then investigated further by grouping them according to the assumed payload-range capability goals set for the derivatives. The findings from this are as follows:
 - *Goal type 1*: keep capacity constant/increase range. This was found to be a common goal and changes are usually focused on increasing fuel capacity by incorporating additional fuel tanks. This is sometimes accompanied by aerodynamic, propulsion, and structural modifications to handle the increase in weights. The average development time appeared to be relatively short for this goal type.
 - *Goal type 2*: increase both payload and range. This was also found

to be a common goal but involved much more extensive modifications, especially stretched fuselages, large increases in wing area, changes to the undercarriage, and others. This is the group in which ‘leaps’ were found the most often and had the highest average development time. The upgrades falling in this group appear to be unplanned and is perhaps where this research can contribute the most.

- *Goal type 3*: decrease payload capacity/increase or decrease range. Relatively few modifications were found, apart from the obvious shrink in fuselage length. The average development time was found to be rather long for this group, however.
 - *Goal type 4*: increase capacity, reduce range (simple stretches). Airframe modifications for this type were also usually minor (insertion of fuselage plugs and local structural enforcements) and development times tended to be short.
 - *Goal type 5*: increase capacity, drastically reduce range. This is where a ‘regional’ variant of a high capacity aircraft is produced and involves decreasing maximum take-off weight while increasing strength to increase fatigue life for short-haul, high capacity operations.
- (h) Maximum changes in salient design parameters were identified and documented in Table 3.23. These were used as constraints for the work presented in Chapter 4.
- (i) The studies confirmed that evolving designs to different configurations was successfully implemented across the Boeing 707, 727, 737, and 757 families.
4. It was found that, given a suitable knowledge representation for two aircraft, a means could be devised to determine automatically which pairs of airframe subcomponents across these two aircraft should be compared to estimate similarity. Specifically, it was shown that such knowledge of the airframes could be represented in the form of an object model and a design structure matrix (DSM). A technique and associated algorithm were developed to parse pairs of these object model and DSM combinations, which provided component pairs to be compared as output.
5. Airframe commonality could be predicted automatically across airframes of dissimilar configurations (see Chapter 4). Specifically, it was found that determining fuselage and nacelle commonality could be posed as a longest common subsequence problem and it was shown how an algorithm developed

during the research to solve this problem was successfully applied to estimate commonality. Additionally, it was determined that a suitable way in which wing commonality could be estimated was to employ a simple multi-objective optimisation problem. This involves determining the span length of a tip-extension and/or root insert to increase the wingspan, which would minimise the chord-length increase. For empennage surfaces, it was shown how commonality could be determined by first estimating the sizes of tip-extension and/or root-insert. Finally, it was shown how the commonality across the undercarriage of two airframes could be estimated by comparing the undercarriage layout and length.

6. It was found through tests, case studies, and interviewing experts that the commonality prediction techniques produce expected results and were useful, not only for estimating redesign effort, but also because it enabled one to obtain suggestions for change rules from the evolvability exploration study, rather than having to pre-specify these.
7. The existing set-based design criteria of ‘interval dominance’, ‘maximality’, and ‘E-admissibility’ could be extended easily to enable systematic reduction of the evolvability design space. This was demonstrated through a test case.
8. It was demonstrated that E-admissibility is not always appropriate as a criterion to eliminate near-future options. This is because the uncertainty regarding which far-future option to pursue would normally be unresolved at the time of deciding which near-future aircraft should be developed. If near-future options were eliminated based on the E-admissibility criterion, a situation might arise in which the baseline has less design reuse potential than the eliminated designs. However, there are exceptions, specifically if the far-future design is planned as a simple derivative, *before* deciding on which baseline to choose.
9. A new set-based design strategy developed during the research, called ‘constrain and union’, was also found to be effective in filtering out inferior design options, thereby allowing designers to reduce the design space swiftly and systematically, while not converging prematurely to too few options.

7.4 Research Contributions

In addition to the above findings, the specific contributions to knowledge and engineering practice resulting from this research are as follows:

1. A computational framework to assess similarity automatically across airframes with dissimilar configurations was proposed and tested. This framework could be considered an architecture for a knowledge-based engineering

application and consists of the following enablers:

- (a) Knowledge representation – a generic aircraft object model and design structure matrix (DSM) was proposed. The object model contains salient information on each constituent aircraft component, whereas the DSM stores information on how the components are attached to each other in a compact form.
- (b) An airframe partitioning module. This is an algorithm that parses the DSMs of two input airframes and provides as output pairs of components from these airframes that should be compared in detail to determine similarity.
- (c) An airframe commonality predictor. For this module, techniques and concepts from mathematics and computer science (the longest common subsequence problem and associated algorithms), multi-objective optimisation, and product platforming were adapted to predict commonality automatically across input airframes, based on information obtained from the object models and DSMs of the airframes.

This extension and compilation of existing techniques from different fields of endeavour (mainly mathematics, computer science, optimisation, and aircraft conceptual design) to develop techniques for a new application (airframe similarity assessment) constitute the academic contribution. Practically, they enable the designer to reduce the computational cost involved in evolvability studies during conceptual design and to obtain suggestions for redesign change rules from their studies, rather than having to pre-specify these. As discussed, these benefits enable more efficient and effective exploration of evolvability.

2. Computational techniques and associated algorithms to enable swift and systematic filtering of the evolvability design space were proposed and tested successfully. To develop these, criteria and techniques from the emerging field of set-based design were extended to the evolvability exploration problem. These techniques are not particular to aircraft evolvability and could be used for studying evolvability in other complex systems. They could also be extended easily for other types of changeability studies. These techniques therefore constitute a contribution to the fields of systems engineering, engineering design, and aircraft design.

7.5 Future Work

As mentioned throughout the thesis, there are a number of limitations associated with the proposed techniques. These could be addressed in future work to improve upon the work presented in this text. These avenues for future research are summarised below.

1. A priority for future work should be to address the potential performance error obtained on the far-future design options when sampling the far-future design space separately from the baseline design space. This would be vital to ensuring that meaningful results are obtained from an evolvability study. More studies, such as was performed in Chapter 1.2, need to be conducted to obtain an improved understanding of how large this error could be. Ideally, methods to quantify the relationship between the commonality achieved and the potential performance error should be developed. Methods from the field of uncertainty quantification and management could also prove useful in predicting and managing these errors.
2. The scope of the similarity assessment techniques could be extended to other engineering systems. The first step in this direction could involve extending these techniques to military (where evolvability is always a major concern) and general aviation aeroplanes. This should be reasonably simple to do, but more case studies would have to be conducted on how evolvability is implemented in these types of aircraft. Other engineering systems to consider could include space vehicles, helicopters, automobiles, and maritime systems.
3. Similarly, the scope could be widened to develop techniques for similarity assessment across subsystems. Δ DSMs [85] could prove promising for such a study. A particularly interesting challenge would be to estimate the overall airframe-subsystem similarity.
4. Although the focus in this research was mainly on similarity because of similar geometry, much of the benefit of evolvability usually also derives from reusing processes (quality control, manufacturing, operations, supply-chain, and so forth). Future work should include considering how these could be incorporated.
5. Although the commonality prediction techniques are fast, they are not fast enough to significantly speed up many simpler aircraft conceptual design studies. Particularly, at their current stage of development, their usefulness is limited to studies in which i) it is first endeavoured to reduce the size of design space before doing similarity assessment, or ii) where execution time per design point is longer, such in more detailed, physics-based design studies (which are becoming more common). Future work should therefore focus on

- ways of expediting commonality prediction. The effects of discretising the fuselage and wing on execution time should be investigated in more depth. Such a study will also have to consider the potential effects on the accuracy of the commonality predicted. The algorithms themselves could possibly be improved as well. The algorithm developed to solve the longest common subsequence, for example, is based on a relatively basic existing algorithm, and more advanced algorithms exist that could solve the LCS faster. These should be investigated to determine whether they could be applied for the current type of study. Other means of assessing similarity in geometries could also be investigated to determine whether they could speed up the process.
6. Developing a simple distance measure (in metric space) might be a promising avenue for developing a faster technique for similarity assessment. The author has investigated this to some extent but, although this work appears promising, it was deemed not mature enough to be included in this thesis.
 7. The set-based techniques produced could be extended to other types of changeability studies. They appear to be especially promising for product family studies, where the ‘timeframe-tradespaces’ could simply be replaced by ‘user-tradespaces’.

References

- [1] M. Kingsley-Jones, “Farnborough first news: The race to rewire the Airbus A380,” *Flight International*, p. 18, 2006.
- [2] C. S. Tang, J. D. Zimmerman, and J. I. Nelson, “Managing new product development and supply chain risks: The Boeing 787 case,” *Supply Chain Forum: An International Journal*, vol. 10, no. 2, pp. 74–86, 2009.
- [3] O. L. de Weck, A. M. Ross, and D. H. Rhodes, “Investigating relationships and semantic sets amongst system lifecycle properties (ilities),” in *Proceedings of the third international engineering systems symposium CESUN*, (Delft University of Technology, Delft, The Netherlands), pp. 1–12, 2012.
- [4] E. Fricke and A. P. Schulz, “Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle,” *Systems Engineering*, vol. 8, no. 4, pp. 342–359, 2005.
- [5] A. M. Ross, D. H. Rhodes, and D. E. Hastings, “Defining system changeability: Reconciling flexibility, adaptability, scalability, and robustness for maintaining system lifecycle value,” in *17th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2007 - Systems Engineering: Key to Intelligent Enterprises*, vol. 2, (San Diego, CA, USA), pp. 735–749, 2007.
- [6] E. T. Ryan, D. R. Jacques, and J. M. Colombi, “An ontological framework for clarifying flexibility-related terminology via literature survey,” *Systems Engineering*, vol. 16, no. 1, pp. 99–110, 2013.
- [7] D. Lim, *A systematic approach to design for lifelong aircraft evolution*. PhD thesis, Georgia Institute of Technology, 2009.
- [8] Boeing, “737 MAX gallery.” <http://www.boeing.com/commercial/737max/#/gallery>. Accessed: 2018-02-20.
- [9] D. P. Raymer, *Aircraft Design: A Conceptual Approach*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 5 ed., 2012.
- [10] D. G. Ullman, *The Mechanical Design Process*. New York, NY: McGraw-Hill, 4 ed., 2010.

- [11] L. A. McCullers, "Aircraft configuration optimization including optimized flight profiles," in *Proceedings of the Symposium on Recent Experiences in Multidisciplinary Analysis and Optimization, NASA CP 2327, April 24–26*, (Hampton, VA), 1984.
- [12] W. Graham, C. Hall, and M. Morales, "The potential of future aircraft technology for noise and pollutant emissions reduction," *Transport Policy*, vol. 34, pp. 36–51, 2014.
- [13] NASA, "Down to earth future aircraft —image gallery." <https://www.nasa.gov/content/down-to-earth-future-aircraft-0>. Accessed: 2018-02-22.
- [14] M. E. Fitzgerald and A. M. Ross, "Mitigating contextual uncertainties with valuable changeability analysis in the multi-epoch domain.," in *Systems Conference (SysCon), 2012 IEEE International*, (Vancouver), pp. 1–8, 2012.
- [15] A. Riaz, *A Set-Based Approach to Passenger Aircraft Family Design*. PhD, Cranfield University, 2015.
- [16] A. Riaz, M. D. Guenov, and A. Molina-Cristobal, "Set-based approach to passenger aircraft family design," *Journal of Aircraft*, vol. 54, no. 1, pp. 310–326, 2016.
- [17] D. K. Sobek, A. C. Ward, and J. K. Liker, "Toyota's principles of set-based concurrent engineering," *Sloan management review*, vol. 40, no. 2, p. 67, 1999.
- [18] L. T. M. Blessing and A. Chakrabarti, *DRM, a design research methodology*. London: Springer-Verlag, 2009.
- [19] EASA, *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes*. Amendment 20, European Aviation Safety Agency, Effective 24 August 2017, 2017.
- [20] FAA, *Federal Aviation Regulations, Part 25 – Airworthiness Standards: Transport Category Airplanes*. Washington, DC, USA: US Federal Aviation Administration, 2009.
- [21] Accreditation Board for Engineering and Technology, "Criteria for accrediting engineering programs, 2018–2019." <http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2018-2019/#definitions>, 2018. Accessed: 2018-06-07.
- [22] A. J. Keane and P. B. Nair, *Computational approaches for aerospace design*. Chichester: Wiley, 2005.
- [23] J. P. Fielding, *Introduction to aircraft design*. New York: Cambridge University Press, 2017.
- [24] E. Torenbeek, *Synthesis of Subsonic Airplane Design*. Delft University Press, Delft, the Netherlands and Kluwer Academic Publishers, Dordrecht, the Netherlands, 1982.
- [25] E. Torenbeek, *Advanced aircraft design: Conceptual design, technology and optimization of subsonic civil airplanes*. United Kingdom: John Wiley & Sons, 2013.
- [26] L. R. Jenkinson, P. Simpkin, and D. Rhodes, *Civil jet aircraft design*. London: Arnold, 1999.

- [27] J. R. R. A. Martins and A. B. Lambe, “Multidisciplinary design optimization: A survey of architectures,” *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, 2013.
- [28] M. Nunez, V. C. Datta, A. Molina-Cristobal, M. D. Guenov, and A. Riaz, “Enabling exploration in the conceptual design and optimisation of complex systems,” *Journal of Engineering Design*, vol. 23, no. 10–11, pp. 849–872, 2012.
- [29] B. M. Kulfan, “Universal parametric geometry representation method,” *Journal of Aircraft*, vol. 45, no. 1, pp. 142–158, 2008.
- [30] B. M. Kulfan, “Recent extensions and applications of the ‘CST’ universal parametric geometry representation method,” *The Aeronautical Journal (1968)*, vol. 114, no. 1153, p. 157–176, 2010.
- [31] M. Padulo and M. D. Guenov, “A methodological perspective on computational engineering design under uncertainty,” in *ECCOMAS 2012 - European Congress on Computational Methods in Applied Sciences and Engineering, e-Book Full Papers*, pp. 7509–7528, 2012.
- [32] O. L. de Weck and C. M. Eckert, “A classification of uncertainty for early product and system design,” 2007.
- [33] M. D. Guenov, X. Chen, A. Molina-Cristóbal, A. Riaz, A. S. van Heerden, and M. Padulo, “Margin allocation and tradeoff in complex systems design and optimization,” *AIAA Journal*, pp. 1–16, 2018.
- [34] E. Kang, E. Jackson, and W. Schulte, “An approach for effective design space exploration,” in *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems*, pp. 33–54, Berlin, Heidelberg: Springer, 2011.
- [35] M. D. Guenov, M. Nunez, A. Molina-Cristóbal, V. C. Datta, and A. Riaz, “AirCADia – an interactive tool for the composition and exploration of aircraft computational studies at early design stage,” in *29th Congress of the International Council of the Aeronautical Sciences, ICAS 2014*, (St. Petersburg, Russia), pp. 1–12, 2014.
- [36] D. C. Montgomery, *Design and analysis of experiments*. New York, NY: John Wiley & Sons, 7 ed., 2009.
- [37] M. Cavazzuti, *Optimization methods: From theory to design scientific and technological aspects in mechanics*. Berlin and Heidelberg: Springer Science & Business Media, 2012.
- [38] G. la Rocca, *Knowledge based engineering techniques to support aircraft design and optimization*. PhD thesis, TU Delft, Delft University of Technology, 2011.
- [39] G. la Rocca, “Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design,” *Advanced engineering informatics*, vol. 26, no. 2, pp. 159–179, 2012.
- [40] D. Auty, “Object oriented programming systems and frame representations, an investigation of programming paradigms,” Tech. Rep. NASA 186084, National Aeronautics and Space Administration, 1988.
- [41] MIT Engineering Systems Division, “Strategic report,” tech. rep., Massachusetts Institute of Technology, 2008.
- [42] E. Fricke, B. Gebhard, H. Negele, and E. Igenbergs, “Coping with changes: Causes, findings, and strategies,” *Systems Engineering*, vol. 3, no. 4, pp. 169–179, 2000.

- [43] S. Wenzel, T. Bauch, E. Fricke, and H. Negele, "Concurrent Engineering and more... A Systematic Approach to Successful Product Development," in *Proceedings of the 7th International Symposium of INCOSE*, (Los Angeles, CA, U.S.A.), pp. 662–669, Wiley Online Library, 1997.
- [44] A. P. Schulz and E. Fricke, "Incorporating flexibility, agility, robustness, and adaptability within the design of integrated systems – key to success?," in *Proceedings of the 18th Digital Avionics Systems Conference*, (St Louis, MO, U.S.A.), pp. 1.A.2–1–8, 1999.
- [45] E. F. Colombo, G. Cascini, and O. L. de Weck, "Classification of change-related ilities based on a literature review of engineering changes," *Journal of Integrated Design and Process Science*, vol. 20, no. 4, pp. 3–23, 2016.
- [46] C. Eckert, P. J. Clarkson, and W. Zanker, "Change and customisation in complex engineering domains," *Research in Engineering Design*, vol. 15, no. 1, pp. 1–21, 2004.
- [47] P. J. Clarkson, C. Simons, and C. Eckert, "Predicting change propagation in complex design," *Journal of Mechanical Design, Transactions of the ASME*, vol. 126, no. 5, pp. 788–797, 2004.
- [48] R. Keller, T. Eger, C. M. Eckert, and P. J. Clarkson, "Visualising change propagation," in *Proceedings ICED 05, the 15th International Conference on Engineering Design*, vol. DS 35, (Melbourne, Australia), pp. 1–12, 2005.
- [49] M. Giffin, O. de Weck, G. Bounova, R. Keller, C. Eckert, and P. J. Clarkson, "Change propagation analysis in complex technical systems," *Journal of Mechanical Design*, vol. 131, no. 8, pp. 81001–810014, 2009.
- [50] D. C. Wynn, N. H. M. Caldwell, and P. J. Clarkson, "Predicting change propagation in complex design workflows," *Journal of Mechanical Design, Transactions of the ASME*, vol. 136, no. 8, pp. 1–13, 2014.
- [51] A. M. Ross and D. E. Hastings, "Assessing changeability in aerospace systems architecting and design using dynamic multi-attribute tradespace exploration," in *Collection of Technical Papers – AIAA Space 2006 Conference*, vol. 1, (San Jose, CA, U.S.A.), pp. 551–568, 2006.
- [52] N. B. Shah, L. Viscito, J. Wilds, A. M. Ross, and D. Hastings, "Quantifying flexibility for architecting changeable systems," in *6th Conference on Systems Engineering Research*, (Los Angeles, CA, U.S.A.), pp. 1–13, 2008.
- [53] E. Fricke, *Der Änderungsprozeß als Grundlage einer nutzerzentrierten Systementwicklung. (The Change Process as a Basis for User-Centered System Development)*. PhD thesis, Institute of Astronautics, Technical University of Munich, Munich, 1999.
- [54] B. Mekdeci, A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Pliability and Viable Systems: Maintaining Value Under Changing Conditions," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2014.
- [55] R. De Neufville and S. Scholtes, *Flexibility in engineering design*. Cambridge, Massachusetts and London, England: The MIT Press, 2011.
- [56] L. Viscito and A. Ross, "Quantifying Flexibility in Tradespace Exploration: Value Weighted Filtered Outdegree," in *AIAA Space 2009 Conference and Expo*, (Pasadena, CA, U.S.A.), pp. 1–9, 2009.

- [57] J. M. Wilds, *A Methodology for Identifying Flexible Design Opportunities*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2008.
- [58] D. F. Wyatt, D. C. Wynn, and P. J. Clarkson, “A computational method to support product architecture design,” in *ASME International Mechanical Engineering Congress and Exposition, Proceedings*, vol. 4, (Lake Buena Vista, FL, USA), pp. 195–208, 2009.
- [59] E. C. Y. Koh, N. H. M. Caldwell, and P. J. Clarkson, “A technique to assess the changeability of complex engineering systems,” *Journal of Engineering Design*, vol. 24, no. 7, pp. 477–498, 2013.
- [60] Z. Pirmoradi, G. G. Wang, and T. W. Simpson, “A review of recent literature in product family design and platform-based product development,” in *Advances in product family and product platform design* (T. W. Simpson, R. J. Jiao, Z. Siddique, and K. Hölttä-Otto, eds.), pp. 1–46, New York: Springer, 2014.
- [61] M. H. Meyer and A. P. Lehnerd, *The power of product platforms*. New York, NY: Free Press, 1997.
- [62] T. W. Simpson, Z. Siddique, and J. R. Jiao, “Platform-based product family development,” in *Product platform and product family design*, pp. 1–15, New York, NY: Springer, 2006.
- [63] T. W. Simpson, J. Jiao, Z. Siddique, and K. Hölttä-Otto, *Advances in product family and product platform design*. New York: Springer, 2014.
- [64] J. Jiao, T. W. Simpson, and Z. Siddique, “Product family design and platform-based product development: A state-of-the-art review,” *Journal of Intelligent Manufacturing*, vol. 18, no. 1, pp. 5–29, 2007.
- [65] E. S. Suh, O. L. de Weck, and D. Chang, “Flexible product platforms: Framework and case study,” *Research in Engineering Design*, vol. 18, no. 2, pp. 67–89, 2007.
- [66] H. A. ElMaraghy, “Changing and evolving products and systems – models and enablers,” in *Changeable and Reconfigurable Manufacturing Systems* (H. A. ElMaraghy, ed.), pp. 25–45, London: Springer, 2009.
- [67] L. L. Zhang, “A literature review on multitype platforming and framework for future research,” *International Journal of Production Economics*, vol. 168, pp. 1–12, 2015.
- [68] B. G. Cameron and E. F. Crawley, “Crafting platform strategy based on anticipated benefits and costs,” in *Advances in Product Family and Product Platform Design* (T. W. Simpson, R. J. Jiao, Z. Siddique, and K. Hölttä-Otto, eds.), pp. 49–70, New York: Springer, 2014.
- [69] R. Whitford, *Evolution of the airliner*. Crowood Press Ltd, 2007.
- [70] American Society of Mechanical Engineers, “Wright Flyer III.” <https://www.asme.org/about-asme/who-we-are/engineering-history/landmarks/224-wright-flyer-iii>, 2018. Accessed: 2018-06-14.
- [71] R. J. Francillon, *McDonnell Douglas Aircraft since 1920*. Oxford: Alden Press, 1988.

- [72] J. Young, R. Anderson, and R. Yurkovich, "A description of the F/A-18E/F design and design process," in *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Multidisciplinary Analysis Optimization Conferences, American Institute of Aeronautics and Astronautics, Sep 1998.
- [73] D. Long and S. Ferguson, "A case study of evolvability and excess on the B-52 Stratofortress and F/A-18 Hornet," in *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V004T05A026–V004T05A026, American Society of Mechanical Engineers, 2017.
- [74] D. Kellari, E. F. Crawley, and B. G. Cameron, "Architectural decisions in commercial aircraft from the DC-3 to the 787," *Journal of Aircraft*, vol. 55, no. 2, pp. 792–804, 2017.
- [75] R. H. Liebeck, "Design of the blended wing body subsonic transport," *Journal of Aircraft*, vol. 41, no. 1, pp. 10–25, 2004.
- [76] D. Scholz, "Design aspects of passenger box wing aircraft," in *3rd Symposium on Collaboration in Aircraft Design, 19–20 September*, (Linköping, Sweden), 2013.
- [77] P. Zhuravlev and V. Zhuravlev, "Significance of modifications for development of passenger airplanes," *Aircraft Engineering and Aerospace Technology*, vol. 84, no. 3, pp. 172–180, 2012.
- [78] J. Beesemyer, *Empirically Characterizing Evolvability and Changeability in Engineering Systems*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA., 2010.
- [79] J. C. Beesemyer, D. O. Fulcoy, A. Ross, and D. H. Rhodes, "Developing methods to design for evolvability: Research approach and preliminary design principles," in *9th Conference on Systems Engineering Research*, (Redondo Beach, CA), 2011.
- [80] M. A. Cardin, "Enabling flexibility in engineering systems: A taxonomy of procedures and a design framework," *Journal of Mechanical Design, Transactions of the ASME*, vol. 136, no. 1, pp. 1–14, 2014.
- [81] A. Aliakbargolkar, *A framework for space systems architecting under stakeholder objectives ambiguity*. PhD thesis, Massachusetts Institute of Technology, 2012.
- [82] A. Golkar and E. F. Crawley, "A framework for space systems architecture under stakeholder objectives ambiguity," *Systems Engineering*, vol. 17, no. 4, pp. 479–502, 2014.
- [83] K. G. Schwartz and D. N. Mavris, "Linking technology readiness to uncertainty reduction," in *30th Congress of the International Council of the Aeronautical Sciences, ICAS 2016*, 2016.
- [84] D. Knoll, A. Golkar, and O. de Weck, "A concurrent design approach for model-based technology roadmapping," in *Systems Conference (SysCon), 2018 Annual IEEE International*, pp. 1–6, IEEE, 2018.
- [85] O. L. de Weck, "On the role of DSM in designing systems and products for changeability," in *DSM 2007: Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.*, 2007.
- [86] R. Smaling and O. de Weck, "Assessing risks and opportunities of technology infusion in system design," *Systems Engineering*, vol. 10, no. 1, pp. 1–25, 2007.

- [87] E. S. Suh, M. R. Furst, K. J. Mihalyov, and O. de Weck, "Technology infusion for complex systems: A framework and case study," *Systems Engineering*, vol. 13, no. 2, pp. 186–203, 2010.
- [88] S. Mathews, V. Datar, and B. Johnson, "A practical method for valuing real options: The Boeing approach," *Journal of Applied Corporate Finance*, vol. 19, no. 2, pp. 95–104, 2007.
- [89] P. Banerjee and O. L. de Weck, "Flexibility strategy—valuing flexible product options," in *INCOSE/ICSE conference on synergy between systems engineering and project management*, (Las Vegas, NV), pp. 1–8, Citeseer, 2004.
- [90] S. K. Moon and T. W. Simpson, "Platform valuation for product family design," in *Advances in Product Family and Product Platform Design*, pp. 179–198, Springer, 2014.
- [91] A. Engel and Y. Reich, "Architecting systems for optimal lifetime adaptability," in *Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies*, (Seoul, South Korea), Aug 19–22 2013.
- [92] F. Biery and M. A. Lorell, "Preplanned product improvement and other modification strategies: Lessons from past aircraft modification programs," Tech. Rep. N-1794-AF, RAND Corporation, Santa Monica, CA, 1981.
- [93] U.S. Department of Defense, "Department of Defense Instruction No. 5000.2." <https://www.acq.osd.mil/dpap/Docs/new/5000.2%2005-12-06.pdf>, 2003. Accessed: 2018-06-21.
- [94] N. P. Suh, *Axiomatic design*. Oxford University Press, 2001.
- [95] A. H. Tilstra, P. B. Backlund, C. C. Seepersad, and K. L. Wood, "Principles for designing products with flexibility for future evolution," *International Journal of Mass Customisation*, vol. 5, no. 1, pp. 22–54, 2015.
- [96] S. White and S. Ferguson, "Exploring architecture selection and system evolvability," in *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V02BT03A044–V02BT03A044, American Society of Mechanical Engineers, 2017.
- [97] J. D. Allen, C. A. Mattson, and S. M. Ferguson, "Evaluation of system evolvability based on usable excess," *Journal of Mechanical Design*, vol. 138, no. 9, p. 091101, 2016.
- [98] M. W. P. Tackett, C. A. Mattson, and S. M. Ferguson, "A model for quantifying system evolvability based on excess and capacity," *Journal of Mechanical Design*, vol. 136, no. 5, pp. 51002–51011, 2014.
- [99] S. D. Eppinger and T. R. Browning, *Design Structure Matrix Methods and Applications*. Cambridge, MA, USA: MIT Press, 2012.
- [100] E. C. Y. Koh, N. H. M. Caldwell, and P. J. Clarkson, "A method to assess the effects of engineering change propagation," *Research in Engineering Design*, vol. 23, no. 4, pp. 329–351, 2012.
- [101] J. Luo, "A simulation-based method to evaluate the impact of product architecture on product evolvability," *Research in Engineering Design*, vol. 26, no. 4, pp. 355–371, 2015.

- [102] R. C. Boas, *Commonality in Complex Product Families: Implications of Divergence and Lifecycle Offsets*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, U.S.A., 2008.
- [103] M. D. Johnson and R. E. Kirchain, “Developing and assessing commonality metrics for product families,” in *Advances in Product Family and Product Platform Design*, pp. 473–502, Springer, 2014.
- [104] J. Markish, *Valuation techniques for commercial aircraft program design*. Master’s thesis, Massachusetts Institute of Technology, 2002.
- [105] J. Watson, J. Allen, C. A. Mattson, and S. M. Ferguson, “Maximizing the value of system evolvability under uncertainty,” in *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, p. 2646, 2015.
- [106] M. R. Silver and O. L. de Weck, “Time-expanded decision networks: A framework for designing evolvable complex systems,” *Systems Engineering*, vol. 10, no. 2, pp. 167–186, 2007.
- [107] A. Babajide, R. de Neufville, and M.-A. Cardin, “Integrated Method for Designing Valuable Flexibility in Oil Development Projects,” *SPE Projects, Facilities & Construction*, vol. 4, no. 2, pp. 3–12, 2009.
- [108] G. A. Hazelrigg, “A Framework for Decision-Based Engineering Design,” *Journal of Mechanical Design*, vol. 120, no. 4, pp. 653–658, 1998.
- [109] A. Olewnik, T. Brauen, S. Ferguson, and K. Lewis, “A Framework for Flexible Systems and Its Implementation in Multiattribute Decision Making,” *Journal of Mechanical Design*, vol. 126, no. 3, pp. 412–419, 2003.
- [110] D. W. Lim and D. N. Mavris, “A design methodology for lifelong aircraft evolution,” in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, vol. 4, (Orlando, FL), pp. 3305–3323, 2010.
- [111] A. M. Ross, D. E. Hastings, J. M. Warmkessel, and N. P. Diller, “Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design,” *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 20–28, 2004.
- [112] R. L. Keeney and H. Raiffa, *Decisions with Multiple Objectives*. Cambridge, United Kingdom: Cambridge University Press, 1993.
- [113] D. L. Thurston, “Real and Misconceived Limitations to Decision Based Design With Utility Analysis,” *Journal of Mechanical Design*, vol. 123, no. 2, pp. 176–182, 1999.
- [114] A. M. Ross and D. E. Hastings, “The tradespace exploration paradigm,” in *15th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2005*, vol. 2, (Rochester, NY, USA), pp. 1706–1718, 2005.
- [115] A. Ross, H. McManus, D. Rhodes, and D. Hastings, “Revisiting the tradespace exploration paradigm: Structuring the exploration process,” in *AIAA SPACE 2010 Conference & Exposition*, 2010.
- [116] M. E. Fitzgerald and A. M. Ross, “Sustaining lifecycle value: Valuable changeability analysis with era simulation,” in *Systems Conference (SysCon), 2012 IEEE International*, (Vancouver), pp. 1–7, 2012.
- [117] A. M. Ross, *Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration*. PhD thesis, Massachusetts Institute of Technology, 2006.

- [118] A. M. Ross and D. H. Rhodes, "Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis," in *INCOSE International symposium*, (Utrecht, The Netherlands), pp. 1186–1201, 2008.
- [119] C. J. Roberts, M. G. Richards, A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Scenario planning in dynamic multi-attribute tradespace exploration," in *2009 3rd Annual IEEE Systems Conference*, pp. 366–371, 2009.
- [120] D. O. Fulcoly, *A Normative Approach to Designing for Evolvability: Methods and Metrics for Considering Evolvability in Systems Engineering*. Master's thesis, Massachusetts Institute of Technology, 2012.
- [121] D. O. Fulcoly, A. M. Ross, and D. H. Rhodes, "Evaluating system change options and timing using the epoch syncopation framework," *Procedia Computer Science*, vol. 8, pp. 22–30, 2012.
- [122] H. L. McManus, M. G. Richards, A. M. Ross, and D. E. Hastings, "A framework for incorporating "ilities" in tradespace studies," in *A Collection of Technical Papers – AIAA Space 2007 Conference*, vol. 1, pp. 941–954, 2007.
- [123] J. E. Derleth, *Multi-Attribute Tradespace Exploration and its Application to Evolutionary Acquisition*. Master's thesis, Massachusetts Institute of Technology, 2003.
- [124] A. S. J. van Heerden, M. D. Guenov, A. Molina-Cristóbal, and A. Riaz, "Enhancing the exploration of aircraft changeability during conceptual design," in *30th Congress of the International Council of the Aeronautical Sciences*, September 25–30, (Daejeon, South Korea), 2016.
- [125] R. Khire, J. Wang, T. Bailey, Y. Lin, and T. Simpson, "Product Family Commonality Selection Using Optimization and Interactive Visualization," in *Advances in Product Family and Product Platform Design*, pp. 449–471, New York: Springer, 2014.
- [126] L. Slingerland, A. Bobuk, and T. Simpson, "A product family optimization approach using multidimensional data visualization," in *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, (Fort Worth, Texas), 2010.
- [127] H. H. Roark, *Value centric approach to target system modularization using multi-attribute tradespace exploration and network measures of component modularity*. Master's thesis, Massachusetts Institute of Technology, 2012.
- [128] J. R. Kizer and D. N. Mavris, "Set-based Design Space Exploration Enabled by Dynamic Constraint Analysis," in *29th Congress of the International Council of the Aeronautical Sciences (ICAS)*, (St. Petersburg, Russia), Sep 7–12, 2014.
- [129] R. J. Malak, J. M. Aughenbaugh, and C. J. Paredis, "Multi-attribute utility analysis in set-based conceptual design," *Computer-Aided Design*, vol. 41, no. 3, pp. 214 – 227, 2009.
- [130] H. Junfei, *Flexible Engineering System Design with Multiple Exogenous Uncertainties and Change Propagation*. PhD, National University of Singapore, 2012.
- [131] A. Cardone, S. K. Gupta, and M. Karnik, "A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications," *Journal of Computing and Information Science in Engineering*, vol. 3, no. 2, pp. 109–118, 2003.

- [132] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani, "Three-dimensional shape searching: State-of-the-art review and future trends," *Computer-Aided Design*, vol. 37, no. 5, pp. 509–530, 2005.
- [133] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in *Proceedings of the Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000*, pp. 39–48, 2000.
- [134] B. Bollobás, G. Das, D. Gunopulos, and H. Mannila, "Time-series similarity problems and well-separated geometric sets," in *Proceedings of the thirteenth annual symposium on computational geometry*, pp. 454–456, ACM, 1997.
- [135] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: The MIT Press, 2nd ed., 2001.
- [136] D. J. Singer, N. Doerry, and M. E. Buckley, "What is set-based design?," *Naval Engineers Journal*, vol. 121, no. 4, pp. 31–43, 2009.
- [137] B. M. Kennedy, D. K. Sobek, and M. N. Kennedy, "Reducing rework by applying set-based practices early in the systems engineering process," *Systems Engineering*, vol. 17, no. 3, pp. 278–296, 2014.
- [138] A. Ward, J. K. Liker, J. J. Cristiano, and D. K. Sobek, "The second Toyota paradox: How delaying decisions can make better cars faster," *Sloan management review*, vol. 36, no. 3, p. 43, 1995.
- [139] W. W. Finch and A. C. Ward, "A set-based system for eliminating infeasible designs in engineering problems dominated by uncertainty," in *Proceedings of the 1997 ASME Design Engineering Technical Conferences, Sacramento, CA, Paper No. DETC97/DTM-3886*, 1997.
- [140] S. J. Rekuc, J. M. Aughenbaugh, M. Bruns, and C. J. Paredis, "Eliminating design alternatives based on imprecise information," tech. rep., SAE Technical Paper, 2006.
- [141] P. Walley, *Statistical reasoning with imprecise probabilities*. New York: Chapman and Hall, 1991.
- [142] I. Levi, "On indeterminate probabilities," *The journal of philosophy*, vol. 71, no. 13, pp. 391–418, 1975.
- [143] Boeing, "Our company: General information." <http://www.boeing.com/company/>. Accessed: 2018-03-12.
- [144] R. Redding and B. Yenne, *Boeing: Planemaker to the World*. London, United Kingdom: Bison Books, 1983.
- [145] G. Norris and M. Wagner, *Boeing*. Minneapolis, MN, U.S.A.: Motorbooks International, 1998.
- [146] Boeing, "Model 367-80 – Historical snapshot." <http://www.boeing.com/history/products/model-367-80.page>. Accessed: 2018-03-06.
- [147] J. E. Steiner, "Jet Aviation Development – One Company's Perspective. The Last Forty Years and a Brief Look at the Future," in *Presented at the National Air and Space Museum, Smithsonian Institution, October 1979*, (Seattle, WA, U.S.A.), 1979.
- [148] J. Winchester, *Airlife's Classic Airlines: Boeing 707/720*. United Kingdom: Airlife Publishing Ltd, 2002.

- [149] B. Almojuela, “The development of Boeing’s 367-80.” Airplanes Pacific Northwest AIAA Technical Symposium, 2009.
- [150] Jane’s by IHS Markit, “Boeing 707.”
https://janes.ihs.com/Janes/Display/jau_1412-jau_, 2018.
Accessed: 2018-06-29.
- [151] Boeing, “CAD 3-View Drawings for Airport Planning Purposes.”
http://www.boeing.com/commercial/airports/\3_view.page, 2018.
Accessed: 2018-05-30.
- [152] W. Green, G. Swanborough, and J. Mowinski, *Modern Commercial Aircraft*. London: Salamander Books, 1987.
- [153] Jane’s by IHS Markit, “Boeing 720.”
https://janes.ihs.com/Janes/Display/jau_1413-jau_, 2018.
Accessed: 2018-06-30.
- [154] Boeing, “707/720 commercial transport – Historical snapshot.”
<http://www.boeing.com/history/products/707.page>. Accessed: 2018-03-07.
- [155] Boeing, “727 commercial transport – Historical snapshot.”
<http://www.boeing.com/history/products/727.page>. Accessed: 2018-03-21.
- [156] Jane’s by IHS Markit, “Boeing 727.”
https://janes.ihs.com/Janes/Display/jau_1414-jau_, 2018.
Accessed: 2018-06-29.
- [157] Boeing Commercial Airplanes, “Airplane characteristics for airport planning.”
https://www.boeing.com/commercial/airports/\plan_manuals.page, 2018. Accessed: 2018-06-01.
- [158] A. J. Wright, *Boeing 737*. London: Ian Allan Ltd., 1991.
- [159] Jane’s by IHS Markit, “Boeing 737.”
<https://janes.ihs.com/Janes/Display/jawal202-jawa>, 2018.
Accessed: 2018-06-01.
- [160] Jane’s by IHS Markit, “Boeing 737-200.”
https://janes.ihs.com/Janes/Display/jau_1415-jau_, 2018.
Accessed: 2018-06-29.
- [161] Jane’s by IHS Markit, “Boeing 737-300.”
https://janes.ihs.com/Janes/Display/jau_9138-jau_, 2018.
Accessed: 2018-06-29.
- [162] Jane’s by IHS Markit, “Boeing 737-400.”
https://janes.ihs.com/Janes/Display/jau_9139-jau_, 2018.
Accessed: 2018-06-29.
- [163] Jane’s by IHS Markit, “Boeing 737-500.”
https://janes.ihs.com/Janes/Display/jau_9140-jau_, 2018.
Accessed: 2018-06-29.
- [164] P. Proctor, “Lower costs drive next-generation 737,” *Aviation Week & Space Technology*, p. 68–71, Dec 1996.

- [165] R. Faye, R. Laprete, and M. Winter, “Blended winglets for improved airplane performance,” *Boeing Aero Magazine*, Jan 2002.
- [166] Boeing, “Boeing launches 737 new engine family with commitments for 496 airplanes from five airlines.” <http://boeing.mediaroom.com/2011-08-30-Boeing-Launches-737-New-Engine-Family-with-Commitments-for-496-Airplanes-from-Five-Airlines>, 2011. Accessed: 2018-06-29.
- [167] J. Ostrower, “Boeing designates 737 MAX family,” *FlightGlobal*, 2017. <https://www.flightglobal.com/news/articles/boeing-designates-737-max-family-361428/>. Accessed: 2018-06-29.
- [168] F. George, “Pilot report: Flying the 737-8, Boeing’s new narrowbody breadwinner,” *Aviation Week & Space Technology*, 2017. <http://aviationweek.com/commercial-aviation/pilot-report-flying-737-8-boeing-s-new-narrowbody-breadwinner>. Accessed: 2018-06-29.
- [169] S. Trimble, “Analysis: 737 MAX cutaway and technical description,” *FlightGlobal*, 2017. <https://www.flightglobal.com/news/articles/analysis-737-max-cutaway-and-technical-description-437370/>. Accessed: 2018-06-29.
- [170] G. Norris, “Laminar flow boosts 737 MAX long-range performance,” *Aviation Week & Space Technology*, 2013. <http://aviationweek.com/awin/laminar-flow-boosts-737-max-long-range-performance>. Accessed: 2018-06-29.
- [171] T. Hephher, “Boeing 737 MAX 7 narrowbody jetliner makes maiden flight,” *Reuters*, 2018. <https://www.reuters.com/article/us-boeing-737-max-7/boeing-737-max-7-narrowbody-jetliner-makes-maiden-flight-idUSKCN1GS2CM>. Accessed: 2018-06-29.
- [172] S. Trimble, “Farnborough: Boeing details range-boosting 737-7 redesign,” *FlightGlobal*, 2016. <https://www.flightglobal.com/news/articles/farnborough-boeing-details-range-boosting-737-7-red-427468/>. Accessed: 2018-06-29.
- [173] G. Polek, “Boeing upscales 737 MAX 7,” *Aviation International News*, 2016. <https://www.ainonline.com/aviation-news/aerospace/2016-07-12/boeing-upscales-737-max-7>. Accessed: 2018-06-29.
- [174] S. Trimble, “Paris: Boeing launches 737 MAX 10,” *FlightGlobal*, 2017. <https://www.flightglobal.com/news/articles/paris-boeing-launches-737-max-10-438370/>. Accessed: 2018-06-29.
- [175] Boeing, “Boeing launches larger capacity 737 MAX 10 at 2017 Paris air show.” <http://boeing.mediaroom.com/2017-06-19-Boeing-Launches-Larger-Capacity-737-MAX-10-at-2017-Paris-Air-Show>, 2011. Accessed: 2018-06-29.
- [176] Jane’s by IHS Markit, “Boeing 747-200.” https://janes.ihs.com/Janes/Display/jau_1416-jau_, 2018. Accessed: 2018-06-29.
- [177] Jane’s by IHS Markit, “Boeing 747SP.” https://janes.ihs.com/Janes/Display/jau_1418-jau_, 2018. Accessed: 2018-06-29.

- [178] Jane's by IHS Markit, "Boeing 747-400." https://janes.ihs.com/Janes/Display/jau_a146-jau_, 2018. Accessed: 2018-06-29.
- [179] Jane's by IHS Markit, "Boeing 747-8." <https://janes.ihs.com/Janes/Display/jawaa644-jawa>, 2018. Accessed: 2018-06-29.
- [180] P. Gilchrist, *Boeing 747*. London: Ian Allan, 1992.
- [181] C. Collisson, "Re-inventing the legend: The development of the 747-8," in *26th International Congress of the Aeronautical Sciences*, (Anchorage, AK, USA), 2008.
- [182] P. J. Birtles, *Boeing 757/767*. Shepperton, Surrey: Ian Allan Ltd, 1987.
- [183] Jane's by IHS Markit, "Boeing 757-200." https://janes.ihs.com/DefenceEquipment/Display/jau_a059-jau_, 2018. Accessed: 2018-06-29.
- [184] Jane's by IHS Markit, "Boeing 757-300." https://janes.ihs.com/DefenceEquipment/Display/jau_a060-jau_, 2018. Accessed: 2018-06-29.
- [185] Jane's by IHS Markit, "Boeing 767." <https://janes.ihs.com/Janes/Display/jawa1207-jawa>, 2018. Accessed: 2018-06-29.
- [186] H. Queen, "Introducing the 767-400 extended range airplane," *Boeing Aero Magazine*, July 1998.
- [187] G. Norris, "Strategic stretch," *Flight International*, pp. 449–471, 25–31 August 1999.
- [188] G. Norris and M. Wagner, *Boeing 777*. St. Paul, Minnesota: Motorbooks International, 1996.
- [189] G. Norris and M. Wagner, *Modern Boeing jetliners*. Osceola, WI: MBI Pub. Co., 1999.
- [190] Jane's by IHS Markit, "Boeing 777." <https://janes.ihs.com/Janes/Display/jawa1209-jawa>, 2018. Accessed: 2018-06-29.
- [191] G. Norris, "First flight of 777-200lr kicks off ultra-long-haul test programme," *FlightGlobal*, 2005. <https://www.flightglobal.com/news/articles/first-flight-of-777-200lr-kicks-off-ultra-long-haul-195155/>. Accessed: 2018-06-29.
- [192] G. Norris, "Long ranger," *FlightGlobal*, 2003. <https://www.flightglobal.com/news/articles/long-ranger-160884/>. Accessed: 2018-06-29.
- [193] S. Trimble, "Dubai: Boeing formally launches 777X family with 259 commitments," *FlightGlobal*, 2013. <https://www.flightglobal.com/news/articles/dubai-boeing-formally-launches-777x-family-with-259-393138/>. Accessed: 2018-06-29.

- [194] S. Trimble, “Boeing reveals ultra-long-range 777-8x,” *FlightGlobal*, 2013. <https://www.flightglobal.com/news/articles/boeing-reveals-ultra-long-range-777-8x-387108/>. Accessed: 2018-06-29.
- [195] Boeing, “777X by design.” <http://www.boeing.com/commercial/777x/by-design#/featured>, 2018. Accessed: 2018-06-29.
- [196] G. Norris and J. Anselmo, “777x production investments bolster Boeing,” *Aviation Week & Space Technology*, 2016. <http://aviationweek.com/commercial-aviation/777x-production-investments-bolster-boeing>. Accessed: 2018-06-29.
- [197] V. Bhaskara and C. Sloan, “Boeing 777x program perched precariously despite strong execution,” *Airways*, 2017. <https://airwaysmag.com/airlines/boeing-777x-trouble-strong-execution/>. Accessed: 2018-06-29.
- [198] Jane’s by IHS Markit, “Boeing 787.” <https://janes.ihs.com/Janes/Display/jawa5814-jawa>, 2018. Accessed: 2018-06-29.
- [199] M. Kingsley-Jones, “Analysis: Why Boeing’s 787-9 is more than just a stretch,” *FlightGlobal*, 2014. <https://www.flightglobal.com/news/articles/analysis-why-boeings-787-9-is-more-than-just-a-str-401110/>. Accessed: 2018-06-29.
- [200] G. Norris, “Simplicity is vital to Boeing 787-10 execution,” *Aviation Week & Space Technology*, 2016. <http://aviationweek.com/commercial-aviation/simplicity-vital-boeing-787-10-execution>. Accessed: 2018-06-29.
- [201] S. Trimble, “Analysis: Boeing 787-10 technical description and cutaway,” *FlightGlobal*, 2018. <https://www.flightglobal.com/news/articles/analysis-boeing-787-10-technical-description-and-cu-446856/>. Accessed: 2018-06-29.
- [202] Jane’s by IHS Markit, “Boeing (Douglas) DC-9.” https://janes.ihs.com/Janes/Display/jau_1428-jau_, 2018. Accessed: 2018-06-29.
- [203] Jane’s by IHS Markit, “Boeing (McDonnell Douglas) MD-80 Series.” https://janes.ihs.com/Janes/Display/jau_9144-jau_, 2018. Accessed: 2018-06-29.
- [204] Jane’s by IHS Markit, “Boeing (McDonnell Douglas) MD-90.” https://janes.ihs.com/Janes/Display/jau_9145-jau_, 2018. Accessed: 2018-06-29.
- [205] Boeing, “717/MD-95 commercial transport – Historical snapshot.” <http://www.boeing.com/history/products/717-md-95.page>. Accessed: 2018-03-06.
- [206] Jane’s by IHS Markit, “Boeing 717.” https://janes.ihs.com/Janes/Display/jau_a092-jau_, 2018. Accessed: 2018-06-29.
- [207] Jane’s by IHS Markit, “Boeing (McDonnell Douglas) DC-10.” https://janes.ihs.com/Janes/Display/jau_1430-jau_, 2018. Accessed: 2018-06-29.

- [208] Jane's by IHS Markit, "Boeing (McDonnell Douglas) MD-11." https://janes.ihs.com/Janes/Display/jau_9143-jau_, 2018. Accessed: 2018-06-29.
- [209] J. Bailey and J. Marsden, "Md-11 enters the fray," *Flight International*, May 1989.
- [210] H. Hopkins, "Masterfully digital 11," *Flight International*, October 1990.
- [211] L. Marriott, *McDonnell Douglas MD-11*. Shepperton, Surrey: Ian Allan Ltd., 1992.
- [212] B. Gunston, *Airbus*. London: Osprey Publishing Company, 1988.
- [213] Jane's by IHS Markit, "Airbus A300." https://janes.ihs.com/Janes/Display/jau_a097-jau_, 2018. Accessed: 2018-06-30.
- [214] A. J. Wright, *Airbus*. Shepperton, Surrey: Ian Allan Ltd., 1984.
- [215] Jane's by IHS Markit, "Airbus A310." https://janes.ihs.com/Janes/Display/jau_a005-jau_, 2018. Accessed: 2018-06-30.
- [216] Jane's by IHS Markit, "Airbus A320." https://janes.ihs.com/Janes/Display/jawa0416-jawa_, 2018. Accessed: 2018-05-24.
- [217] Jane's by IHS Markit, "Airbus A318." https://janes.ihs.com/Janes/Display/jawa0417-jawa_, 2018. Accessed: 2018-06-30.
- [218] Jane's by IHS Markit, "Airbus A319." https://janes.ihs.com/Janes/Display/jawa0418-jawa_, 2018. Accessed: 2018-06-30.
- [219] Jane's by IHS Markit, "Airbus A321." https://janes.ihs.com/Janes/Display/jawa0419-jawa_, 2018. Accessed: 2018-06-30.
- [220] Boeing, "First A321LR successfully completes maiden flight." <http://www.airbus.com/newsroom/press-releases/en/2018/01/first-a321lr-successfully-completes-maiden-flight.html>, 2018. Accessed: 2018-06-30.
- [221] P. K. C. Rudolph, "High-lift systems on commercial subsonic airliners," Tech. Rep. NASA Contractor Report 4746, National Aeronautics and Space Administration, Ames Research Center, California 94035-1000, 1996.
- [222] M. Kingsley-Jones, "Airbus twin still peaking," *Flight International*, pp. 32–39, November 2017.
- [223] Jane's by IHS Markit, "Airbus A330." https://janes.ihs.com/Janes/Display/jawa0422-jawa_, 2018. Accessed: 2018-06-30.
- [224] Airbus, "Airport Operations – AutoCAD 3 view aircraft drawings." <http://www.airbus.com/aircraft/support-services/airport-operations-and-technical-data/autocad-3-view-aircraft-drawings.html>, 2018. Accessed: 2018-05-30.

- [225] G. Norris and M. Wagner, *Airbus A340 and A330*. St. Paul, MN, USA: Motorbooks International, 2001.
- [226] Jane's by IHS Markit, "Airbus A340." <https://janes.ihs.com/Janes/Display/jawa0421-jawa>, 2018. Accessed: 2018-06-30.
- [227] Airbus, "Aircraft characteristics: Airport operations and technical data." <http://www.airbus.com/aircraft/support-services/airport-operations-and-technical-data/aircraft-characteristics.html>, 2018. Accessed: 2018-06-01.
- [228] Airbus Airport Operations, "A330neo Airport Operations Brochure – Issue 1," 2016.
- [229] D. Kaminski-Morrow, "Farnborough: Airbus outlines A330neo engineering demands," *FlightGlobal*, 2014. <https://www.flightglobal.com/news/articles/farnborough-airbus-outlines-a330neo-engineering-dem-401664/>. Accessed: 2018-06-30.
- [230] Jane's by IHS Markit, "Airbus A350 XWB." <https://janes.ihs.com/Janes/Display/jawaa054-jawa>, 2018. Accessed: 2018-06-30.
- [231] D. Kaminski-Morrow, "Picture: First a350-900ulr starts flight-test campaign," *FlightGlobal*, 2018. <https://www.flightglobal.com/news/articles/picture-first-a350-900ulr-starts-flight-test-campai-447920/?cmpid=SOC|Twitter|Flightglobal|sf187551548|sf187551548&sfid=701w0000000uP3H#sf187551548>.
- [232] D. Kaminski-Morrow, "Picture: Airbus rolls out first a350-900ulr," *FlightGlobal*, 2018. <https://www.flightglobal.com/news/articles/picture-airbus-rolls-out-first-a350-900ulr-446286/>.
- [233] M. Kingsley-Jones, "Airbus opts for larger wing on a350-1000 through trailing edge extension," *FlightGlobal*, 2010. <https://www.flightglobal.com/news/articles/airbus-opts-for-larger-wing-on-a350-1000-through-tra-341004/>.
- [234] Jane's by IHS Markit, "Airbus A380." <https://janes.ihs.com/Janes/Display/jawa5306-jawa>, 2018. Accessed: 2018-06-30.
- [235] J. Flottau, "Airbus details A380-plus modifications," *FlightGlobal*, 2017. <http://aviationweek.com/paris-air-show-2017/airbus-details-a380-plus-modifications>. Accessed: 2018-06-30.
- [236] B. N. Nield, "An overview of the boeing 777 high lift aerodynamic design," *The Aeronautical Journal*, vol. 99, no. 989, pp. 361–371, 1995.
- [237] G. Norris, "Boeing 777 – unveiling a design classic (1990)," *Aviation Week & Space Technology*, 2015. <http://aviationweek.com/blog/boeing-777-unveiling-design-classic-1990>. Accessed: 2018-07-01.
- [238] Pacelab, "Pacelab APD, PACE Aerospace Engineering and Information Technology GmbH," 2016.
- [239] L. Cavagna, S. Ricci, and L. Travaglini, "NeoCASS: an integrated tool for structural sizing, aeroelastic analysis and MDO at conceptual design level," *Progress in Aerospace Sciences*, vol. 47, no. 8, pp. 621–635, 2011.

- [240] Boeing, “Current market outlook: 2017–2036.” <http://www.boeing.com/resources/boeingdotcom/commercial/market/current-market-outlook-2017/assets/downloads/2017-cmo-6-19.pdf>, 2017. Accessed: 2018-05-17.
- [241] R. Del Rosario, J. Koudelka, R. Wahls, and N. Madavan, “NASA subsonic fixed wing project overview,” in *2012 NASA Fundamental Aeronautics Program Technical Conference, Renaissance Cleveland Hotel, Cleveland*, vol. 14, 2012.
- [242] S. W. Ashcraft, A. S. Padron, K. A. Pascioni, G. W. Stout Jr, and D. L. Huff, “Review of propulsion technologies for n+3 subsonic vehicle concepts. nasa/tm–2011-217239,” tech. rep., NASA, Glenn Research Center, Cleveland, OH 44135, 2011.
- [243] Airbus, “Getting to grips with aircraft noise,” tech. rep., Airbus Customer Services, Blagnac, France, 2011.
- [244] U.S. Department of Transportation, Federal Aviation Administration, “Electronic Code of Federal Regulations, Title 14.I.C-Part 36 — Noise Standards: Aircraft Type and Airworthiness Certification,” 2018. https://www.ecfr.gov/cgi-bin/retrieveECFR?gp=&SID=848a71308ceac7a9cc4798db08de4dac&n=14y1.0.1.3.19&r=PART&ty=HTML#se14.1.36_11. Accessed: 2018-05-17.
- [245] N. Dickson, “Local air quality and ICAO engine emissions standards (presentation).” https://www.icao.int/Meetings/EnvironmentalWorkshops/Documents/2014-Kenya/4-1_LAQ-Technology_notes.pdf. Accessed: 2018-05-18.
- [246] CFM International, “LEAP Overview.” https://www.cfmaeroengines.com/wp-content/uploads/2017/09/Brochure_LEAPfiches_2017.pdf. Accessed: 2018-05-18.
- [247] S. Bruner, S. Baber, C. Harris, N. Caldwell, P. Keding, K. Rahrig, L. Pho, and R. Wlezian, “NASA N+3 subsonic fixed wing silent efficient low-emissions commercial transport (SELECT) vehicle study. Revision A,” tech. rep., National Aeronautics and Space Administration, Glenn Research Center, Cleveland, OH 44135, 2010.
- [248] M. D. Guenov, M. Nunez, A. Molina-Cristóbal, V. Sripawadkul, V. Datta, and A. Riaz, “Composition, management, and exploration of computational studies at early design stage,” *Computational Intelligence in Aerospace Sciences*, pp. 415–460, 2014.
- [249] U. Michel, “Correlation of aircraft certification noise levels EPNL with controlling physical parameters,” in *19th AIAA/CEAS Aeroacoustics Conference*, (Berlin, Germany), May 27–29, 2013.
- [250] GasTurb GmbH, “GasTurb.” <http://www.gasturb.de/index.html>, 2018. Accessed: 2018-05-21.
- [251] Southwest Research Institute, “NPSS – Numerical Propulsion System Simulation.” <https://www.swri.org/consortia/numerical-propulsion-system-simulation-npss>, 2018. Accessed: 2018-05-21.
- [252] L. Lopes and C. Burley, “Design of the next generation aircraft noise prediction program: ANOPP2,” in *17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference)*, p. 2854, 2011.

- [253] M. Sensmeier and J. Samareh, “A Study of Vehicle Structural Layouts in Post-WWII Aircraft,” in *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Structures, Structural Dynamics, and Materials and Co-located Conferences, American Institute of Aeronautics and Astronautics, Apr 2004.
- [254] A. Seitz, J. Bijewitz, S. Kaiser, and G. Wortmann, “Conceptual investigation of a propulsive fuselage aircraft layout,” *Aircraft Engineering and Aerospace Technology: An International Journal*, vol. 86, no. 6, pp. 464–472, 2014.
- [255] Airbus, “Airbus concept plane (image).” https://airbus-h.assetsadobe2.com/is/image/content/dam/corporate-topics/innovation-concepts/Airbus-Concept_plane-side_view_right.jpg?wid=3626&fit=constrain, 2010. Accessed: 2018-05-31.
- [256] Airbus S.A.S. Customer Services, “Airbus A320: Aircraft characteristics – airport and maintenance planning. Revision No. 34 (1 February 2018).” http://www.airbus.com/content/dam/corporate-topics/publications/backgrounders/techdata/aircraft_characteristics/Airbus-Commercial-Aircraft-AC-A320-Feb18.pdf, 2018. Accessed: 2018-05-22.
- [257] European Aviation Safety Agency, “Type certificate data sheet for the LEAP-1A and LEAP-1C series engines (CFM International SA), TCDS No: E.110, Issue: 06.” <https://www.easa.europa.eu/sites/default/files/dfu/EASA%20E110%20TCDS%20Issue%206%20LEAP-1A-1C.pdf>, 2017. Accessed: 2018-05-24.
- [258] D. Moule, “Easyjet noise reduction update (presentation).” <http://www2.westsussex.gov.uk/ds/cttee/gat/gat260117i12.pdf>, 2017. Accessed: 2018-05-24.
- [259] M. D. Guynn, J. J. Berton, K. L. Fisher, W. J. Haller, M. T. Tong, and D. R. Thurman, “Refined exploration of turbofan design options for an advanced single-aisle transport,” Tech. Rep. NASA/TM-2011-216883, National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia 23681-2199, 2011.
- [260] M. D. Guynn, J. J. Berton, W. J. Haller, E. S. Hendricks, and M. T. Tong, “Performance and environmental assessment of an advanced aircraft with open rotor propulsion,” Tech. Rep. NASA/TM-2012-217772, National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia 23681-2199, 2012.

APPENDIX A

Civil Transport Aircraft Evolvability Data

In this appendix, the data that was used to create the plots and figures in Chapter 3, Sections 3.5 and 3.6 are provided. Tables A.1, A.2, and A.3, show each aircraft surveyed, along with the launch (or first order date), the certification (or entry-into-service) date, and the estimated development time (in months, between the launch and EIS dates). Table A.4 lists all the references consulted for each of the aircraft types considered and apply across all the tables in this index. Table A.5 provides a list of changes considered for each of the major components. In this table, each change has an ID, which is used in Tables A.6 to A.13 to see if took place in a certain aircraft (1) or not (0).

Table A.1: Boeing aircraft surveyed.

ID	Type	Baseline	Launch/Order	Certification/ EIS	Development time [months]
1	367-80	0	01/04/1952	15/07/1954	27
2	707-120	1	13/10/1955	23/09/1958	35
3	707-320	2		15/07/1959	
4	720	2	01/07/1957	30/06/1960	35
5	707-120B	2		01/03/1961	
6	720B	4		03/03/1961	
7	707-320B	3		31/05/1962	
8	727-100	0	05/12/1960	24/12/1963	36
9	727-200	8		30/11/1967	
10	737-100	0	12/05/1964	15/12/1967	43
11	737-200	10		21/12/1967	815
12	747-100	0	13/04/1966	30/12/1969	44
13	747-100B SR	12			
14	747-200B	12		23/12/1970	
15	737-200Adv	11	12/05/1964	01/05/1971	83
16	727-200Adv	9	12/05/1971	14/06/1972	13
17	747SP	12	01/09/1973	04/02/1976	29
18	767-200	0	14/07/1978	30/07/1982	48
19	757-200	0	13/08/1978	21/12/1982	52
20	747-300	14	12/06/1980	07/03/1983	32
21	767-200ER	18	16/12/1982	01/03/1984	14
22	737-300	15	01/09/1980	14/11/1984	50
23	767-300	21	29/09/1983	22/09/1986	35
24	767-300ER	23	21/07/1984	20/01/1988	41
25	737-400	22	01/06/1986	02/09/1988	27
26	747-400	20	22/10/1985	10/01/1989	38
27	747-400ER	26			
28	737-500	15	20/05/1987	18/01/1993	67
29	777-200	0	29/10/1990	19/04/1995	53
30	777-200ER	29			
31	737-700 winglets	22	17/11/1993	07/11/1997	47
32	737-800	25	05/09/1994	13/03/1998	42
33	777-300	30	26/06/1995	04/05/1998	34
34	737-600	28	15/03/1995	18/08/1998	41
35	757-300	19	02/09/1996	27/01/1999	28
36	767-400ER	24	01/03/1997	20/07/2000	40
37	737-900	32	10/11/1997	17/04/2001	41
38	777-300ER	33	29/02/2000	16/03/2004	48
39	777-200LR	30	01/03/2003	01/02/2006	35
40	737-700ER	31	31/01/2006	16/02/2007	12
41	737-900ER	37	18/07/2005	20/04/2007	21
42	787-8	0	26/04/2004	26/08/2011	88
43	747-8I	27	14/11/2005	14/12/2011	73
44	787-9	42	01/07/2010	13/06/2014	47
45	737 MAX 8	32	30/08/2011	08/03/2017	66
46	787-10	44	18/06/2013	22/01/2018	55

Table A.2: McDonnell Douglas aircraft surveyed.

ID	Type	Baseline	Launch/Order	Certification/ EIS	Development time [months]
1	DC-8-10	0	07/06/1955	31/08/1959	50
2	DC-8-20	1		19/01/1960	
3	DC-8-30	2		01/02/1960	
4	DC-8-55	3		01/05/1961	
5	DC-9-15	0	08/04/1963	23/11/1965	31
6	DC-8-61	4	01/04/1965	02/09/1966	17
7	DC-9-32	5	01/11/1964	27/01/1967	26
8	DC-8-62	4		27/04/1967	
9	DC-9-41	7	01/04/1966	27/02/1968	22
10	DC-9-21	5	01/11/1966	11/12/1968	25
11	DC-10-10	0	19/02/1968	29/07/1971	41
12	DC-10-30	11		21/11/1972	874
13	DC-9-51	9	05/07/1973	01/08/1975	24
14	MD-81	13	01/10/1977	26/08/1980	34
15	MD-82	14	16/04/1979	31/07/1981	27
16	DC-8-63	6		01/04/1982	
17	MD-83	14	31/01/1983	01/02/1985	24
18	MD-87	17	01/01/1985	21/10/1987	33
19	MD-88	15	16/04/1979	31/07/1981	27
20	MD-11	12	30/12/1986	08/11/1990	46
21	MD-11ER	20		08/11/1990	
22	MD-90-30	19	14/11/1989	16/11/1994	60
23	MD-95/B717	22	01/10/1995	01/09/1999	47

Table A.3: Airbus aircraft surveyed.

ID	Type	Baseline	Launch/Order	Certification/ EIS	Development time [months]
1	A300B2-101	0	29/05/1969	15/03/1974	57
2	A300B2-320	1		15/03/1974	
3	A300B4-203	2		26/03/1975	
4	A310-200	3	07/07/1978	11/03/1983	56
5	A300-600	3	01/12/1980	01/06/1984	42
6	A310-300	4		01/04/1986	
7	A300-600R	5		10/03/1988	
8	A320-200	0	23/03/1984	08/11/1988	55
9	A340-200	7	05/06/1987	22/12/1992	66
10	A340-300	7	05/06/1987	22/12/1992	66
11	A330-300	7		21/10/1993	
12	A321-100	8	22/05/1989	17/12/1993	54
13	A319-100	8	10/06/1993	10/04/1996	34
14	A321-200	12	12/04/1995	24/04/1997	24
15	A330-200	11	24/11/1995	31/03/1998	28
16	A340-600	10	08/12/1997	29/04/2002	52
17	A340-500	10		03/12/2002	
18	A318	13	26/04/1999	23/05/2003	48
19	A380-800	0	08/12/1999	12/12/2006	84
20	A321-200 Shark- let	14	15/11/2009	01/12/2012	36
21	A320-200 Shark- let	8	15/11/2009	01/12/2012	36
22	A319-100 Shark- let	13	15/11/2009	01/12/2012	36
23	A350-900	0	17/07/2006	30/09/2014	98
24	A330-300 HGV	11	01/11/2012	01/04/2015	29
25	A330-200 HGV	15		08/09/2015	
26	A320neo	21	01/12/2010	24/11/2015	59
27	A330-300 Re- gional	11	01/09/2013	18/08/2016	35
28	A321neo	20		15/12/2016	
29	A350-1000	23	17/07/2006	21/11/2017	136

Table A.4: References consulted.

Type (all versions)	References
Boeing 367-80	[144], [145], [147], [149], [148], [150], [151]
Boeing 707 and 720	[144], [145], [147], [149], [148], [150], [152], [157], [153], [151]
Boeing 717/MD-95	[205], [157], [151], [206]
Boeing 727	[156], [144], [157], [151]
Boeing 737 Originals	[144], [145], [159], [147], [152], [157], [151]
Boeing 747	[144], [145], [176], [147], [152], [157], [151], [174], [177], [178], [179], [181]
Boeing 767	[144], [182], [145], [152], [185], [186], [187]
Boeing 757	[183], [184], [152], [182], [157], [151]
Boeing 737 Classics	[145], [159], [157], [158], [151]
Boeing 777	[188], [189], [190], [191], [192], [193], [195], [194], [157], [151], [196], [197]
Boeing 737 NG	[145], [159], [157], [151], [164]
Boeing 787	[198], [199], [157], [151], [200], [201]
Douglas DC-8	[157], [151], [152], [71]
McDonnell Douglas DC-9	[157], [151], [152], [71], [202]
McDonnell Douglas MD-80	[157], [151], [152], [203]
McDonnell Douglas DC-10	[157], [151], [152], [203], [207]
McDonnell Douglas MD-11	[157], [151], [208], [209], [210], [211]
Airbus A300	[152], [212], [213], [227], [224]
Airbus A310	[152], [212], [213], [214], [215], [227], [224]
Airbus A320 family	[152], [216], [219], [218], [217], [220], [221], [227], [224]
Airbus A330	[227], [224], [222], [225], [223], [228], [229]
Airbus A340	[227], [224], [226], [225]
Airbus A350	[227], [224], [230], [231], [232], [233]
Airbus A380	[227], [224], [235], [234]

Table A.5: Change descriptions.

Component	Change ID	Description
Fuselage	A	Bulkheads moved/modified
	B	Fuselage frames inserted
	C	Fuselage frames removed
	D	Relofting of segments
	E	Underfloor tanks added
Wing	A	New
	B	Wingtip extension/ revised wingtip
	C	Wing root insert
	D	Other span increase plug
	E	Modified/added wingtip device
	F	New/revised trailing-edge flaps
	G	New/revised leading-edge flaps
	H	Inter-spar chord-increase
	I	Inter-spar chord-decrease
	J	Relofting of aerofoil
	K	Other planform changes
	L	Centre fuel tank added
	M	Modified/new wing-body fairing
	N	Incidence angle changed
	O	Control surface removed
Horizontal stabiliser	A	New
	B	Tip extension
	C	Root-insert
	D	Larger or modified elevator
	E	Tailplane tanks added/higher fuel capacity in tanks
	F	Scaled-up
	G	Scaled-down
Vertical stabiliser	A	New
	B	Tip extension
	C	Root-insert
	D	Other planform change
	E	Scaled-up
	F	Larger or modified rudder
Undercarriage	A	New main
	B	Extended nose gear
	C	Extended main undercarriage
	D	Undercarriage modified (mechanism, trucks, brakes, or wheels)
Engines	A	New

Table A.6: Changes for Goal Type 1: wing and fuselage.

	ID	Fuselage					Wing														
		A	B	C	D	E	A	B	C	D	E	F	G	H	I	J	K	M	N	O	P
707-120B	5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
720B	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
707-320B	7	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0
747-200B	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
737-200Adv	15	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
727-200Adv	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
767-200ER	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
767-300ER	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
747-400	26	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
737-500	28	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0
777-200ER	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
737-700 winglets	31	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0
737-600	34	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0
777-300ER	38	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
777-200LR	39	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
737-900ER	41	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
737 MAX 8	45	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
DC-8-20	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-8-30	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
DC-9-21	10	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
DC-10-30	12	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-82	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-8-63	16	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-83	17	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-88	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A300B2-320	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A300B4-203	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A310-300	6	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
A300-600R	7	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
A321-200	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
A330-300 HGW	24	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0
A330-200 HGW	25	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
A320neo	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A321neo	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
No. of aircraft	34	1	0	0	1	6	0	8	0	0	6	6	7	2	0	0	2	10	3	0	0
% with change		0.03	0.00	0.00	0.03	0.18	0.00	0.24	0.00	0.00	0.18	0.18	0.21	0.06	0.00	0.00	0.06	0.29	0.09	0.00	0.00

Table A.7: Changes for Goal Type 1: other components.

	ID	Horizontal stabiliser							Vertical stabiliser						Undercarriage				Prop
		A	B	C	D	E	F	G	A	B	C	D	E	F	A	B	C	D	A
707-120B	5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
720B	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
707-320B	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
747-200B	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
737-200Adv	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
727-200Adv	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
767-200ER	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
767-300ER	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
747-400	26	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
737-500	28	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1
777-200ER	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
737-700 winglets	31	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0
737-600	34	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0
777-300ER	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
777-200LR	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
737-900ER	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
737 MAX 8	45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
DC-8-20	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-8-30	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-9-21	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-10-30	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
MD-82	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-8-63	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-83	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-88	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A300B2-320	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A300B4-203	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A310-300	6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
A300-600R	7	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
A321-200	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A330-300 HGW	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A330-200 HGW	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A320neo	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
A321neo	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
No. of aircraft	34	0	3	0	3	3	0	0	0	0	2	1	1	0	0	2	0	4	8
% with change		0.00	0.09	0.00	0.09	0.09	0.00	0.00	0.00	0.00	0.06	0.03	0.03	0.00	0.00	0.06	0.00	0.12	0.24

Table A.8: Changes for Goal Type 2: wing and fuselage.

	ID	Fuselage					Wing														
		A	B	C	D	E	A	B	C	D	E	F	G	H	I	J	K	M	N	O	P
707-120	2	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
707-320	3	0	1	0	0	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0	0
737-200	11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
737-300	22	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0
737-800	32	0	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0
747-81	43	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0
787-9	44	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-8-55	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-9-32	7	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
DC-8-62	8	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-9-51	13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-81	14	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0
MD-11	20	0	1	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
A300-600	5	1	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1
A340-200	9	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A340-300	10	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A330-300	11	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A321-100	12	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
A340-600	16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A340-500	17	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0
A350-1000	29	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
No. of aircraft	21	2	19	0	3	1	5	8	2	1	2	8	5	2	0	1	1	1	1	0	1
% with change		0.10	0.90	0.00	0.14	0.05	0.24	0.38	0.10	0.05	0.10	0.38	0.24	0.10	0.00	0.05	0.05	0.05	0.05	0.00	0.05

Table A.9: Changes for Goal Type 2: other components.

	ID	Horizontal stabiliser								Vertical stabiliser						Undercarriage				Prop A
		A	B	C	D	E	F	G		A	B	C	D	E	F	A	B	C	D	
707-120	2	0	0	0	0	0	1	0		1	0	0	0	1	0	1	0	0	0	0
707-320	3	0	0	0	0	0	1	0		0	1	0	0	0	0	0	0	0	0	0
737-200	11	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
737-300	22	0	1	0	1	0	0	0		0	0	0	1	0	0	0	1	0	0	1
737-800	32	0	1	0	1	0	0	0		0	0	1	0	0	0	0	0	0	1	0
747-81	43	0	0	0	0	1	0	0		0	0	0	0	0	1	0	0	0	1	1
787-9	44	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0
DC-8-55	4	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	1
DC-9-32	7	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
DC-8-62	8	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
DC-9-51	13	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
MD-81	14	0	0	1	0	0	0	0		0	0	0	1	0	0	0	0	0	0	1
MD-11	20	1	0	0	0	1	0	0		0	0	0	0	0	0	0	0	0	0	0
A300-600	5	0	0	0	0	0	0	1		0	0	0	0	0	1	0	0	0	0	0
A340-200	9	0	0	0	0	0	0	0		0	0	0	0	0	0	1	0	0	0	1
A340-300	10	0	0	0	0	0	0	0		0	0	0	0	0	0	1	0	0	0	1
A330-300	11	0	0	0	0	0	0	0		0	0	0	0	0	0	1	0	0	0	1
A321-100	12	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0
A340-600	16	0	0	0	0	0	1	0		0	1	0	0	0	1	0	0	0	0	1
A340-500	17	0	0	0	1	0	1	0		0	1	0	0	0	1	0	0	0	1	1
A350-1000	29	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0
No. of aircraft	21	1	2	1	3	2	4	1		1	3	1	2	1	4	4	1	0	5	9
% with change		0.05	0.10	0.05	0.14	0.10	0.19	0.05		0.05	0.14	0.05	0.10	0.05	0.19	0.19	0.05	0.00	0.24	0.43

Table A.10: Changes for Goal Type 3: wing and fuselage.

	ID	Fuselage					Wing														
		A	B	C	D	E	A	B	C	D	E	F	G	H	I	J	K	M	N	O	P
Boeing 720	4	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
A318	18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
747SP	17	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
MD-87	18	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-95/B717	23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
A310-200	4	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
A319-100	13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A330-200	15	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
No. of aircraft	8	1	0	8	3	1	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1
% with change		0.13	0.00	1.00	0.38	0.13	0.00	0.00	0.00	0.00	0.00	0.13	0.13	0.00	0.13	0.00	0.13	0.13	0.13	0.13	0.13

Table A.11: Changes for Goal Type 3: other components.

	ID	Horizontal stabiliser								Vertical stabiliser						Undercarriage				Prop
		A	B	C	D	E	F	G		A	B	C	D	E	F	A	B	C	D	A
Boeing 720	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A318	18	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
747SP	17	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
MD-87	18	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
MD-95/B717	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
A310-200	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
A319-100	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A330-200	15	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
No. of aircraft	8	0	1	0	0	0	0	1	0	0	4	0	0	0	3	0	0	0	1	1
% with change		0.00	0.13	0.00	0.00	0.00	0.00	0.13	0.00	0.50	0.00	0.00	0.00	0.00	0.38	0.00	0.00	0.00	0.13	0.13

Table A.12: Changes for Goal Type 4: wing and fuselage.

	ID	Fuselage					Wing														
		A	B	C	D	E	A	B	C	D	E	F	G	H	I	J	K	M	N	O	P
727-200	9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
747-300	20	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
767-300	23	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
737-400	25	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
777-300	33	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
757-300	35	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
767-400ER	36	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
737-900	37	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
787-10	46	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-8-61	6	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
DC-9-41	9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-90-30	22	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
No. of aircraft	12	0	12	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
% with change		0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.08	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.13: Changes for Goal Type 4: Other components.

	ID	Horizontal stabiliser							Vertical stabiliser						Undercarriage				Prop
		A	B	C	D	E	F	G	A	B	C	D	E	F	A	B	C	D	A
727-200	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
747-300	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
767-300	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
737-400	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
777-300	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
757-300	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
767-400ER	36	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
737-900	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
787-10	46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-8-61	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DC-9-41	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MD-90-30	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
No. of aircraft	12	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
% with change		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.08	0.08

APPENDIX B

MATLAB[®] Script for Example in the Text

B.1 Fuselage Commonality Example

The MATLAB[®] script below is for the example in Section 4.5.1.4.

```
1 %% Fuselage Commonality Prediction Example
2 clc,clear all,close all
3 % Search length limiter:
4 Delta = 3;
5
6 % Discretised x-locations:
7 L1 = 5;
8 L2 = 7;
9
10 X = (0:max(L1,L2))';
11 dx = 1;
12
13 % Pre-assigning non-common segment lists:
14 NCS_1 = [X(1:end-1) X(2:end)];
15 NCS_2 = [X(1:end-1) X(2:end)];
16
17 ind_too_large_1 = find(NCS_1(:,2) > L1);
18 NCS_1(ind_too_large_1,:) = [];
19
20 ind_too_large_2 = find(NCS_2(:,2) > L2);
21 NCS_2(ind_too_large_2,:) = [];
22
23 % Fuselage 1:
24 a11 = [0, 4, 5, 5, 3, 1, 10, 10]; % Fuselage width
25 a12 = [0, 4, 5, 5, 3, 1, 10, 10]; % Fuselage height
26
27 q11 = [0, 0, 0, 1, 0, 0, 10, 10]; % Wing presence
28 q12 = [0, 0, 0, 0, 0, 1, 10, 10]; % HT presence
29 q13 = [0, 0, 0, 0, 0, 0, 10, 10]; % Engine presence
30
31 % Fuselage 2:
32 a21 = [0, 4, 5, 5, 5, 5, 3, 1]; % Fuselage width
33 a22 = [0, 4, 5, 5, 5, 5, 3, 1]; % Fuselage height
34
35 q21 = [0, 0, 0, 0, 1, 0, 0, 0]; % Wing presence
36 q22 = [0, 0, 0, 0, 0, 0, 0, 0]; % HT presence
37 q23 = [0, 0, 0, 0, 0, 0, 1, 0]; % Engine presence
38
39 % Error tolerance:
```

```

40 e1 = 0.2; % Tolerance for difference in fuselage width
41 e2 = 0.2; % Tolerance for difference in fuselage height
42
43 % Begin computing recursive array, R:
44 s = length(X);
45 R = zeros(s,s);
46
47 for k = 2:s
48     for m = max(2,k-Delta):min(s,k+Delta)
49         % First, assign segment to be non-common:
50         R(k,m) = max(R(k-1,m),R(k,m-1));
51
52         if (a11(k) >= a21(m)/(1+e1) && a11(k) <= a21(m)*(1+e1)) && ...
53             (a12(k) >= a22(m)/(1+e2) && a12(k) <= a22(m)*(1+e2))
54
55             if q11(k) == q21(m) && q12(k) == q22(m) && q13(k) == q23(m)
56
57                 if ((a11(k) > a11(k-1) && a21(m) > a21(m-1)) || ...
58                     (a11(k) == a11(k-1) && a21(m) == a21(m-1)) || ...
59                     (a11(k) < a11(k-1) && a21(m) < a21(m-1))) && ...
60                     ((a12(k) > a12(k-1) && a22(m) > a22(m-1)) || ...
61                     (a12(k) == a12(k-1) && a22(m) == a22(m-1)) || ...
62                     (a12(k) < a12(k-1) && a22(m) < a22(m-1)))
63                     % Segment can be considered common.
64                     %(Overwrite previous assignment for R).
65                     R(k,m) = R(k-1,m-1)+1;
66
67                 end
68             end
69         end
70     end
71 end
72
73 % Find LCS (x-locations of common cross-sections):
74 u = 1;
75 while k ~= 1 || m ~= 1
76     if R(k,m) > R(k-1,m) && R(k,m) > R(k,m-1)
77         xp1(u,1) = X(k); %#ok<SAGROW>
78         xp2(u,1) = X(m);
79         k = k-1;
80         m = m-1;
81         u = u+1;
82     elseif R(k,m) == R(k-1,m)
83         k = k-1;
84     else
85         m = m-1;
86     end
87 end
88
89 xp1 = flipud(xp1);
90 xp2 = flipud(xp2);
91
92 % Common segment lists:
93 CS_1 = [xp1-dx xp1];
94 CS_2 = [xp2-dx xp2];
95
96 % Update non-common segment lists:
97 for i = 1:length(CS_1(:,1))
98     indCmn_1 = find(NCS_1(:,1) == CS_1(i,1));
99     NCS_1(indCmn_1,:) = [];
100
101     indCmn_2 = find(NCS_2(:,1) == CS_2(i,1));
102     NCS_2(indCmn_2,:) = [];
103 end
104
105 % Complete segment lists:
106 Count = 1;
107 CS_1_int(1,:) = CS_1(1,:);
108 CS_2_int(1,:) = CS_2(1,:);
109
110 for i = 2:length(CS_1(:,1))
111     if CS_1(i-1,2) == CS_1(i,1) && CS_2(i-1,2) == CS_2(i,1)
112         CS_1_int(Count,2) = CS_1(i,2);
113         CS_2_int(Count,2) = CS_2(i,2);
114     else
115         Count = Count + 1;
116         CS_1_int(Count,1) = CS_1(i,1);
117         CS_2_int(Count,1) = CS_2(i,1);
118     end
119 end

```

```

120
121 CS_1_int(end,2) = CS_1(end,2);
122 CS_2_int(end,2) = CS_2(end,2);
123
124 Count = 1;
125 NCS_1_int(1,:) = NCS_1(1,:);
126 for i = 2:length(NCS_1(:,1))
127     if NCS_1(i-1,2) == NCS_1(i,1)
128         NCS_1_int(Count,2) = NCS_1(i,2);
129     else
130         Count = Count + 1;
131         NCS_1_int(Count,1) = NCS_1(i,1);
132     end
133 end
134
135 Count = 1;
136 NCS_2_int(1,:) = NCS_2(1,:);
137 for i = 2:length(NCS_2(:,1))
138     if NCS_2(i-1,2) == NCS_2(i,1)
139         NCS_2_int(Count,2) = NCS_2(i,2);
140     else
141         Count = Count + 1;
142         NCS_2_int(Count,1) = NCS_2(i,1);
143     end
144 end
145
146 NCS_1_int(end,2) = NCS_1(end,2);
147 NCS_2_int(end,2) = NCS_2(end,2);
148
149 lc_1 = 1:length(CS_1_int(:,1));
150
151 SS_1 = [CS_1_int;NCS_1_int];
152 [SS_1,Ind_memr_1] = sort(SS_1,1);
153
154 SS_2 = [CS_2_int;NCS_2_int];
155 [SS_2,Ind_memr_2] = sort(SS_2,1);
156
157 % Commonality matrix:
158 lss_1 = length(SS_1(:,1));
159 lss_2 = length(SS_2(:,1));
160 BCM = zeros(lss_1,lss_2);
161
162 for i = 1:lss_1
163     for j = 1:lss_2
164         if Ind_memr_1(i,1) == Ind_memr_2(j,1) && ...
165             Ind_memr_1(i,1) <= length(CS_1_int(:,1))
166             BCM(i,j) = 1;
167         end
168     end
169 end
170
171 BCM;

```

Evaluation: Software Implementation

In this Appendix, brief descriptions are provided of the software models employed in the evaluation of this research (Chapter 6).

C.1 Case Study Model Descriptions

C.1.1 Aircraft Sizing: FLOPS

As discussed in Chapter 6, FLOPS is a comprehensive aircraft sizing and performance software. Also, as could be seen in Figure 6.2, it receives several inputs and produces a number of outputs. It has many input parameters (too many to cover here), but most were set to default or, if not specific to the configurations or technologies considered, were set to match the Airbus A320.

C.1.2 Configuration-Technology Combination Selector

This model takes as input the configuration-technology combination ID (AC_ID) and provides the required information for FLOPS, high-lift device, noise, and cost calculation models. This information includes such aspects as wing location (high/mid/low), anhedral or dihedral, wing sweep, tail configuration, engine locations, and the presence of slats.

C.1.3 Control Surface Mass Calculation

This model calculates the mass of the high-lift devices, based on their type (slats or either single, double, or triple-slotted flaps). The work from Rudolph was employed for this purpose [221]. The input was the flap/slat area, which was calculated for each aircraft, based on the input flap/slat chord to wing-chord ratio at fixed percentage locations along the span. The output value overwrites the default FLOPS value, to reflect the difference in mass for different types of high-lift devices.

C.1.4 High-Lift Device Aerodynamics Prediction

This model, based on Raymer [9, pp. 427-429], calculates the lift and drag increments due to the deployment of the high-lift devices. The same geometrical input as for the control surface mass calculation was employed. The output is used in noise calculation and field-length performance (in FLOPS).

C.1.5 Cost

For computing development and manufacturing cost, the model from Markish [104] was employed. The input to this model includes the mass of components or segments of components, the configuration ID, and an indication of whether the component/segment is common to a baseline (only if commonality prediction was already run). The values in Markish's model was adjusted for inflation and reflects the value for 2017. For components that constitute a substantial technological advancement, such as the noise-shielding U-tail, strut-braced wing, or laminar flow wing, the cost was multiplied by a factor of 1.5 to reflect the expected higher investment that would be required to successfully adopt these technologies. The cost of engines or systems were not calculated in this study, as the focus is on airframes. The rest of the assumptions are covered in the main text.

C.1.6 Noise Calculator

As stated in the text, the models from Michel [249] was adopted for calculating flyover, sideline and approach noise. It takes several parameters as input, including geometry and engine specifications. This model had to be calibrated to match predicted values for noise in open rotors. This was done by adjusting the 'tuning' parameters the model. This is a crude approach, but reasonable for the scope of the study under consideration.

C.1.7 Utility Calculator

This model takes the noise, fuel-burn, and fuselage width as input, to calculate the utility of a design, using the equations provided in Section 6.2.3.

C.2 Workflow Calibration

To reasonably simulate the fuel-burn benefits to be obtained with the new engines, the workflow first had to be calibrated using data for an existing aircraft. After this initial calibration, selected parameters in FLOPS could be adjusted such that the promised reduction in fuel-consumption could be matched. Since the study is concerned with possible replacements for an Airbus A320 sized aircraft, this aircraft was selected as the baseline to calibrate the models against. Note that weight-variant WV000 of the A320 was selected for this purpose (see [256, p. 2-2-1]).

To perform the calibration, the FLOPS calibration factors for empty weight, fuel-flow, lift-induced, and lift independent drag were adjusted until the maximum takeoff weight, operating empty weight, and fuel load required for the design mission (payload = 14,250 kg and range = 2,700 nmi [26, Data Set A]) matched the published values for the A320. The results of this calibration study are shown in Table C.1 and, as can be seen, the mass values matched those of the A320 within one percent. The block-fuel calculated by FLOPS for this mission was 15,370 kg.

With the calibrated baseline established, the new propulsion could be incorporated, after which the FLOPS calibration factors could be adjusted such that the resulting reduction in fuel-consumption matched published predictions.

Table C.1: Workflow calibration results

	FLOPS	Published	Percentage difference	Published reference
Maximum takeoff weight [kg]	73,550	73,500	0.07%	[256, p. 2-1-1]
Operational empty weight [kg]	41,354	41,310	0.11%	[256, p. 2-1-1]
Maximum landing weight [kg]	64,720	64,500	0.34%	[256, p. 2-1-1]
Design fuel load [kg]	17,941	17,940	0.01%	[26, Data Set A]

For the LEAP engine, the objective was to match the quoted reduction of 15% in fuel-consumption [216] for the A320neo, as compared with the baseline A320. Note that this value includes the benefits of drag-reducing winglets (‘sharklets’). To this purpose, all airframe and payload masses of the baseline A320 were kept-as is, while the nacelle and engine shape and mass parameters were updated to match published values for these for the LEAP engine (Ref. [257] for mass, and Ref. [256] for dimensions). After this, the FLOPS fuel-flow and drag calibration factors were adjusted until a reduction in mission block-fuel consumed of 14.92% was finally obtained.

For the F3STF, the masses were kept the same again, along with the benefits obtained with winglets for the A320neo. Here a block-fuel reduction of about 10% was assumed, which approximates the gain of 11% (15% - 4% for winglets [258]) in fuel-efficiency obtained with the LEAP engine over the CFM-56 baseline (i.e. it was assumed that a similar generational gain would be obtained). An increase in mass and size for the engine and nacelle was also accounted for by adjusting the appropriate FLOPS parameters for these. These were based on values obtained for engines of similar bypass ratios and application in Guynn et al. [259, p. 8]. The calibration finally resulted in a fuel-burn reduction of 9.57%.

The same was done for the CROR engine, but, in this case, a 12% reduction in block-fuel compared with the F3STF was assumed. This is based on results from Guynn et al. [260], who showed that this is the percentage reduction that could be expected for open-rotors over turbofans of equivalent technology. To account for mass differences, the mass of the geared open-rotor engine studied in Guynn et al. was divided by the ramp weight of the application aircraft, to obtain a factor with which the engine and nacelle masses could be scaled in FLOPS. The length and diameter of the nacelle for this geared open-rotor was also used to update nacelle geometry in FLOPS.

In addition, the empty weight of the airframe had to be increased. This is because CROR configurations require additional strengthening of the airframe to withstand vibration caused by the rotors, as well as noise-insulation in the cabin, amongst other system changes. To do this, the associated mass increase, reported by Guynn et al. in their CROR study [260], was divided by the empty-weight for their baseline aircraft to obtain a percentage increase that could be used to set the weight calibration factors in FLOPS. After applying these changes, the fuel-flow factor in FLOPS was adjusted until an overall reduction in block-fuel consumed of 11.95% was finally obtained (as compared with the F3STF).

C.3 Commonality Prediction Implementation

In this section, the functioning of the MATLAB[®] scripts in which the commonality prediction procedures are implemented (see Chapter 6.3.2) is discussed.

C.3.1 Aircraft Instantiation

This module instantiates the *Aircraft* class (see Section 4.3.1) for a given input aircraft. As can be seen from Figure 6.5, it receives three input data files for a selected aircraft.

The geometry input file is created in AirCADia Geo, a parametric aircraft geometry generation software developed by the Advanced Engineering Design Group at Cranfield University. This software employs the parametric geometry techniques proposed by Kulfan [29] and enables a designer to rapidly create almost any aircraft geometry. The extension of the geometry file is ‘.airpag’, but it is essentially in xml format and is read as such by MATLAB[®].

Mass and cost data is stored in another input file, which is in ‘.mat’ format (i.e. it is a MATLAB[®] data struct). This file contains the mass and cost breakdowns of the selected aircraft. It includes overall weights, such as Maximum Take-off Weight and Operating Empty Weight, as well as component and systems masses. For most of the studies performed here, the data in this file originated from the Aircraft Sizing, Utility and Cost calculator presented in Section 6.2.4.1, but, as will be discussed, for some examples other sources were used.

The DSM input files are also in .mat format and were created manually. This file stores the DSM as a matrix in the form proposed in Section 4.3.2

With these input files, the `ac.inst.m` function performs several tasks. The first is to appropriately combine the information from each of these files and store it in a MATLAB[®] struct that embodies the structure of the aircraft object model presented in Section 4.3.1. It also produces the mass and cost distributions for the fuselage, nacelles, and wings that are required for the detailed commonality prediction.

Because FLOPS does not create mass distributions, these had to be devised using another method. For this purpose, the mass estimation methods from Torenbeek [24, pp. 457-465] (for fuselages) and [25, pp. 319-362] (for wings) were implemented in the script. Using these techniques, the mass contributions of smaller sub-items, such as doors, high-lift devices and other control surfaces, bulkheads, and so forth were calculated first. The assumption was then made that these masses are uniformly distributed with either their length or area and they were added at appropriate locations along the span (for the wings) or distance from the nose (for fuselages). The sum of the masses for these smaller items was then subtracted from the total fuselage or wing masses. For the fuselage, the difference is the shell-weight, which was then distributed according to the cross-sectional area of the fuselage along its length. For the wing, the difference in mass is the basic wing-structure weight and this was distributed across the span according to the distribution shown in [25, p. 343]. For nacelles, it was assumed that the mass distributions simply follow the distribution of cross-sectional area for these components.

The cost distributions were produced by multiplying the mass distribution vectors obtained with the above techniques for each component by the normalised cost parameters from Markish [104]. For the nose and aft segments of the fuselages, the cost was increased such that the cost per unit mass for these segments were 1.5 that of other segments, but with the overall cost remaining the same.

Finally, these distributions are added to the structs which are then saved as .mat files, ready to be used by the airframe partitioning module. The process described above to determine the mass and cost distributions is crude and involved many assumptions.

Dedicated tools that provide distributed masses as part of the sizing process should be employed for actual design exercises, along with more accurate techniques for estimating the cost. However, for the purposes of demonstrating the proposed techniques, the above suffice.

C.3.2 Airframe Partitioning

The object structs produced by the instantiation module are next fed into the aircraft partitioning module, along with the DSM structs. After internally creating category-connection matrices based on the DSMs (see Section 4.4), this module decides which pairs of components should be compared with each other, by employing Algorithm 4.1. Recall that this algorithm produces a comparison matrix for each type of component. ‘Sub-structs’ representing the constituent components are created from the main aircraft struct, which are sent to the detailed commonality prediction scripts, based on the results from the comparison matrices (all wings and all fuselages are always sent for comparison, but for this research it was assumed that there is only one fuselage and one wing per aircraft).

C.3.3 Body Commonality Prediction

The body commonality prediction module script (b_CPM.m) constitutes the software implementation of the techniques proposed in Section 4.5.1. By using the input fuselage or nacelle commonality prediction parameters (tolerances, search limiters, and so forth), along with the fuselage (Fus_a and Fus_b) or nacelle (Nac_a_1, Nac_a_2,..., etc. and Nac_b_1, Nac_b_2,..., etc.) body objects structs, the script first creates the attribute and attached-component sequences as vectors with an associated x -location vector. Linear interpolation is then used to create uniformly discretised sequence vectors (i.e. A_i and Q_t in equations 4.8 and 4.9, respectively) along the x -axis of the component for a given Δx . For the design case study runs, $\Delta x = 0.1$ m for fuselages and $\Delta x = 0.05$ m for nacelles produced acceptable results within a reasonable amount of time. The attributes and component attachment information considered for the studies presented herein are listed below.

- Wing y -location.
- Wing attachment type.
- Empennage surfaces carry-through structures.
- Engine housed in body.
- Nacelle attached directly to body.
- Pylon attachment.
- Undercarriage attachment type.
- CST parameters [29] for the body cross-sections.
- Height and width of cross-section.
- Cross-section reference origin y - and z -deviation (with respect to the cross-section immediately ahead).

The attribute and attached-component sequences of the two bodies are then fed to the ‘Airframe Body Longest Common Subsequence’ solver – a function (called bLCS.m) that incorporates Algorithms 4.2 and 4.3. The implementation of this function is similar to the one shown in Appendix B.1. A tolerance of $\varepsilon = 0.02$ was used for all the non-integer parameters. This value ensured that possible longitudinal errors in the segmentation remained low (below 50 cm). For every new study, it is suggested that the user vary this tolerance to investigate what its effects entail. The ‘search length limiter’ (δ) is calculated for each pair of bodies, by taking the maximum of either the difference between the lengths of the longest and shortest x -vectors of the two bodies, or the rounded-up value

of the length of the longest x -vector divided by 5 (length here refers to the number of entries in the vector). The bLCS script produces as output the segment lists (containing the start and end x -values of each segment) and the corresponding commonality matrices (see the end of Section 4.5.1.3).

The option was also added to this module to predict commonality across bodies with separate top and bottom parametric descriptions, such as would be found with double-bubble fuselages. This simply involves calling the LCS script for each part.

With the common and non-common segments now identified, the mass and cost commonality scores are calculated with Equations 4.3 and 4.5, respectively. Note that the cost commonality score is calculated based on the full development cost for the components. However, as mentioned earlier, the cost model in Markish [104] also provides notional percentage reductions in cost for components that are common with another design (i.e. have been developed as part of a previous design exercise). These cost reduction factors are used by the b_CPM script to provide an adjusted cost value (i.e. the overall predicted cost benefits) to be displayed in the tradespaces when commonality is investigated across designs.

The overall output of the b_CPM script is a struct (called fCmn for fuselages or nCmn for nacelles) containing all the segments, the associated segment mass and cost values, the commonality matrix, and the commonality mass and cost scores.

C.3.4 Wing Commonality Prediction

The wing commonality prediction module was implemented in a script called w_CPM.m. This script receives the wing object structs (Wing_a and Wing_b) and, with the input wing commonality prediction parameters (weighting factors, maximum permitted parameter values, and so forth), it produces a wing commonality struct (called wCmn), using the methods described in Chapter 4.5.2.

The constraints and parameters used for the root-chord insert and tip-extension optimisation (see Equation 4.25) are provided in Table C.2. The weighting factors (w_1 , w_2 , and w_3) chosen reflect that more emphasis is placed on minimising the length of the root-insert and the difference in chord than reducing the overall difference in surface area along the compared span length. The constraints for chord differences are rounded-down values of the maximum changes found for the aircraft surveyed in Chapter 3 (see Table 3.23).

Table C.2: Wing span extension optimisation parameters.

Constraint/parameter	Value
w_1	0.43
w_2	0.33
w_3	0.24
$\Delta \bar{c}_{max}^-$ (root)	−16%
$\Delta \bar{c}_{max}^+$ (root)	16%
$\Delta \bar{c}_{max}^-$ (tip)	−24%
$\Delta \bar{c}_{max}^+$ (tip)	26%

The optimisation is performed by w_CPM as follows (please refer to Figure 4.16 for the meanings of the variable names): the values for the variable Δy (see Equation 4.25) are stored in an array that contains the sequence formed when adding a step value to zero until a maximum value (the positive difference in semispan lengths of the two wings) is reached. This step value is calculated by dividing the difference in semispan lengths

$(y_{tip}^{(l)} - y_{tip}^{(s)})$ by 50. The value of 50 was found to be high enough to keep spanwise errors reasonable. For all the values in the Δy array, the shorter wing and the longer wing (between Δy and $\Delta y + y_{tip}^{(s)}$) are both discretised by dividing them into segments of equal (spanwise) length. This length was calculated by dividing the length of the shorter wing ($y_{tip}^{(s)}$) by 1,000. For this study, the selection of 1,000 kept possible spanwise errors low (below 5 cm), while keeping the computation time acceptable (the user could set this depending on their own needs). With the wings discretised, the value of $f(\Delta y)$ in Equation 4.25 could be calculated for each value of Δy . The value of Δy that produces the minimum value for $f(\Delta y)$ (and which does not violate any of the constraints) is then used as the spanwise length of the root-insert. The tip-extension is subsequently found by subtracting $\Delta y + y_{tip}^{(s)}$ from $y_{tip}^{(l)}$.

After the optimisation, the script calculates the mass and cost commonality scores with the methods presented in Sections 4.5.2.2 to 4.5.2.5. The final output is a struct (wCmn) containing all the wing commonality information in a similar format as that of the body commonality structs (see the previous section).

C.3.5 Empennage Commonality Prediction

The instance structs of the empennage surface pairs (e.g. vertical tail 1 of aircraft a – VT_a.1) are sent to the empennage commonality prediction module (a script called e_CPM.m). If the airframe partitioning module determined that detailed commonality prediction is not appropriate for a particular pair, the commonality values for this pair are set to zero and full development cost applies. Else, the method presented in Section 4.5.3 is executed to calculate the commonality scores and potential cost benefits. The maximum allowable increase and decrease in surface areas (see Equation 4.50) selected for the studies here are shown in Table C.3. The allowable increase values are those found for the aircraft surveyed in Chapter 3 (see Table 3.23). The decrease in area is the allowable decrease in area. In a case where there is a decrease in area and the decrease falls within this allowable difference, the larger surface will be applied to the descendant aircraft and full commonality will apply.

Table C.3: Maximum allowable changes in empennage surface areas.

	Vertical tails	Horizontal tails
$\Delta S_{emp,j}^{(-)}$ (decrease)	-20%	-20%
$\Delta S_{emp,j}^{(+)}$ (increase)	17.41%	27.50%

As with the fuselages and wings, the output of this script is a commonality struct (e-Cmn) which stores the commonality scores and cost reduction.

C.3.6 Undercarriage Commonality Prediction

The undercarriage commonality prediction method, introduced in Chapter 4.5.4, was implemented into a struct called u_CPM.m. The value for n_{uc} in Equation 4.52 was assigned to be 0.5 and the maximum percentage allowable change in length ($\Delta L_{uc}^{(max)}$ in Equation 4.54) was set at 9.37%. This value is the maximum found for the aircraft surveyed in Chapter 3 (Table 3.23).

If the input structs indicate that the undercarriage of the one aircraft is of tricycle arrangement and the other multi-bogey, the centre gear is set as non-common. If both are

of tricycle arrangement, the centre gear of each is investigated to determine if the number of wheels is the same. If indeed, are subjected to the same length difference requirements as other arrangements. As before, the final output of this script is a commonality struct (`uc.Cmn`) containing the commonality and cost-reduction information.

C.3.7 Support Component Commonality Prediction

The final commonality prediction module is for comparing supporting components (struts, pylons, and booms). This module, named `s_CPM.m`, employs the method presented in Chapter 4.5.5 to create a struct containing the commonality scores and cost reduction. Note that only pylons and struts were considered for the studies presented below. For struts, it was specified that they have to have the same length to be considered common (i.e. $\varepsilon_i = 0$ in Equation 4.56). For pylons, if the nacelle was found to be common across the two aircraft, the pylons were set to be common as well.

C.3.8 Commonality Aggregation

The commonality aggregation script (`cmn_agr.m`) simply combines all the results from the commonality prediction modules, as described in Section 4.5.6. It provides a struct (`acCmn`) that contains the overall aircraft mass and cost commonality scores, the component segmentation information, as well as the overall cost reduction predicted for the reuse of the common components for the derivative aircraft.

C.3.9 Commonality Plots

The commonality plots script uses the output struct from the commonality aggregation module, along with the aircraft object instances, to produce geometry renderings of the two aircraft that highlight what components (or segments of components) were found to have commonality. If a component/segment is rendered white (or black for the undercarriage) for the aircraft in one of the plots, it means that there is no commonality across this component/segment and those in the aircraft in the other plot. If, however, there is a coloured component/segment, it means that there is partial or full commonality with a component of similar colour (at relatively the same location) in the plot of the other aircraft.

APPENDIX D

Evaluation Questionnaire



Research Evaluation Questionnaire:

Author: Albert S.J. van Heerden (Full-time PhD student, Cranfield University)

Supervisors: Prof. M.D. Guenov and Dr A. Molina-Cristóbal

Research title: Computational Techniques for Aircraft Evolvability Exploration during Conceptual Design

Department: College of Aeronautics – Advanced Engineering Design Group

Date: 30 January 2018

Introduction

The purpose of this questionnaire is to help me obtain feedback on the usefulness and industrial relevance of my research. For a brief overview of the research, please refer to the summary provided in Appendix C.

Objectives of the evaluation session

The objectives of the session will be as follows:

- To introduce my research to a group of practicing designers.
- To obtain feedback from this group regarding:
 - the validity of my assumptions;
 - the validity of the techniques themselves;
 - the usefulness/relevance of the approach; and
 - how the techniques could be improved.

Format

The session is intended to be completed within two hours. The outline is envisioned to be as follows:

- Introductory presentation (25 min)
- Questions and discussion (30 min)
- Questionnaire (15 min)
- Open questions/further discussion (50 min)

Required from participants

Before the session:

- Kindly read and complete the informed consent form provided in Appendix A, as well as the contact details form in Appendix B.

After the introductory presentation:

- Please ask questions if something was not clear or if you require more information.
- Discussion is most welcome.

When completing the questionnaire:

- Please answer the questions to the best of your knowledge.
- If a question is not clear, please feel free to ask me to elaborate.

After the session:

- Do contact me if you would like to add something to what was already said, or if you would like more information.

Questionnaire

Question 1

Please indicate to what extent you agree or disagree with the following statements by marking the appropriate box.

1. The degree of dissimilarity between two aircraft designs, if measured in an appropriate and consistent manner, would provide a reasonable indication of the possible effort (time and financial expenditure) needed to change any one of these designs into the other.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

2. The distance measure results, shown during the presentation, are reasonable comparative estimates for the redesign effort required.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

3. Similarly, the commonality results are also reasonable estimates.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

4. The commonality prediction metrics would allow me to swiftly identify possible reusable segments or components across different configurations.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

5. The removal of the need to pre-specify transition paths, as discussed, will enable me to reduce the set-up time of my evolvability studies.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

6. The techniques presented will enable me to consider more designs, and specifically more configurations, during my evolvability studies.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

7. Assuming the designs are modelled well and the similarity metrics are accurate, this ability to explore more designs will increase the probability of finding promising design solutions.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

8. The ability to explore more designs will also enable me to increase the level of understanding I can obtain from my evolvability studies.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

9. Increasing my understanding and the probability of finding good designs in this manner will increase the effectiveness of the exploration process, which would help me to improve the decision-making regarding the evolvability of my designs.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

10. Increasing the number of designs and configurations that could be considered also improves the efficiency of the evolvability exploration process, further improving exploration quality.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

11. If implemented in a user-friendly, interactive software tool, the techniques will be useful and add value to the company.

Strongly disagree	Disagree	Neither agree or disagree	Agree	Agree strongly

Question 2

Is there anything that you particularly liked about any of the presented techniques? Which of the techniques do you believe would be the most useful?

Question 3

Is there anything that you did not particularly like about any of the presented techniques?

APPENDIX E

Publications by the Author

The research in this thesis directly led to the following publications by the author:

1. A.S.J. van Heerden, M.D. Guenov, A. Molina-Cristóbal, and A. Riaz, “Automatic Commonality Prediction for Systems with Complex Geometries,” **In preparation:** to be submitted for peer review to the journal *Advanced Engineering Informatics*, 2019.
2. A.S.J. van Heerden, M.D. Guenov, and A. Molina-Cristóbal, “Evolvability and Design Reuse in Civil Transport Aircraft,” **Accepted:** *Progress in the Aerospace Sciences*, 2019.
3. A.S.J. van Heerden, M.D. Guenov, A. Molina-Cristóbal, and A. Riaz and Y. Bile, “Combined Airframe and Subsystems Evolvability Exploration During Conceptual Design,” Presented at the *31st International Council of Aeronautical Sciences (ICAS) Congress*, Belo Horizonte, Brazil, September 9-14, 2018.
4. A.S.J. van Heerden, M.D. Guenov, A. Molina-Cristóbal, and A. Riaz, “Enhancing the Exploration of Aircraft Changeability during Conceptual Design,” Presented at the *30th Congress of the International Council of the Aeronautical Sciences*, Daejeon, Korea, September 25-30, 2016.

The author also co-authored the following related publications, while a research student at Cranfield University:

1. M.D. Guenov, A. Riaz, Y. Bile, A. Molina-Cristóbal and A.S.J. van Heerden, “Information System Support for Aerospace Vehicle Systems Architecting,” Submitted for peer review to the *Journal of Aerospace Information Systems*, 2019.
2. M.D. Guenov, X. Chen, A. Molina-Cristóbal, A. Riaz, A.S.J. van Heerden, and M. Padulo, “Margin Allocation and Trade-off in Complex Systems Design and Optimization,” *AIAA Journal*, pp. 1–16, 2018.

3. A. Molina-Cristóbal, X. Chen, M.D. Guenov, A. Riaz, and A.S.J. van Heerden. “Interactive Uncertainty Allocation and Trade-off at Early-Stage Aircraft Computational Design,” Presented at the *2018 AIAA Non-Deterministic Approaches Conference*, Kissimmee, Florida, 8–12 January, 2018.
4. M.D. Guenov, A. Molina-Cristóbal, V. Voloshin, A. Riaz, A.S.J. van Heerden, S. Sharma, C. Cuiller, and T. Giese, “Aircraft Systems Architecting — a Functional-Logical Domain Perspective,” Presented at the *16th AIAA Aviation Technology, Integration, and Operations Conference*, Washington D.C, 2016.