A modified Beddoes-Leishman dynamic stall model for wind turbine applications

Reinecke, Arne

2018

Reinecke, A. (2018). A modified Beddoes-Leishman dynamic stall model for wind turbine applications. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/74184>

<https://doi.org/10.32657/10356/74184>

A MODIFIED BEDDOES-LEISHMAN
DYNAMIC STALL MODEL
FOR WIND TURBINE APPLICATIONS

ARNE REINECKE

School Of Mechanical And Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2018

Abstract

Dynamic stall is phenomenon that plays a role in many applications such as helicopter or wind turbine aerodynamics. Proper modelling of unsteady lift, drag, and pitching moment characteristics of an airfoil undergoing dynamic stall can help to give more accurate performance predictions, but has turned out to be extremely challenging. The so-called Beddoes-Leishman models – based on indicial formulation with semi-empirical parameters – are considered current state-of-the-art and are incorporated in Blade Element Momentum as well as Vortex Wake Methods in codes for aerodynamic wind turbine analysis. These models contain empirical parameters and time constants that need to be determined manually by comparison to experimental data in a tedious, 'trial & error' fashion.

This thesis investigates the uses of numerical techniques to find optimal values for these parameters. A Evolutionary Strategy is employed optimizing the parameter set and thus, coming up with a parameter set that gives similar, if not better match to the experimental data than default values.

Contents

Nomenclature	15
1 Background	22
1.1 Dynamic Stall	22
1.2 Application Of Dynamic Stall	24
1.3 Modelling Of Dynamic Stall	25
1.4 Motivation For This Work	26
2 Dynamic Stall Model	28
2.1 The AeroDyn Dynamic Stall Model	28
2.1.1 Unsteady Attached Flow (Submodel)	30
Normal Force	31
Chordwise Force	34
Pitching Moment	34
2.1.2 Unsteady Partially Separated Flow (Submodel)	36
Normal Force	38
Chordwise Force	40
Pitching Moment	41
2.1.3 Unsteady Fully Separated Flow (Submodel)	41
Normal Force	43
Chordwise Force	43
Pitching Moment	44
2.2 Experimental Data And Test Cases	44
2.2.1 Transformation Of Loads Into From α Into Time Space . .	46
2.3 Performance Of The AeroDyn Dynamic Stall Model	48
2.3.1 Test Case 101	48
2.3.2 Test Case 102	49
2.3.3 Test Case 103	50
2.3.4 Test Case 104	50
2.3.5 Test Case 105	50
2.3.6 Test Case 106	50
2.3.7 Test Case 107	51
2.3.8 Test Case 108	51
2.3.9 Discussion	51

CONTENTS

2.4	Sensitivity Of Parameters And Time Constants In The NREL Dynamic Stall Model	52
2.4.1	Parameters A_1, A_2, b_1, b_2	52
2.4.2	Time Constant T_f	53
2.4.3	Time Constant T_p	53
2.4.4	Time Constant T_V	53
2.4.5	Time Constant T_{VL}	54
3	Optimization Methods For Finding Optimal Time Constants And Parameters In Beddoes-Leishman Dynamic Stall Models	55
3.1	Optimization – General Remarks	55
3.2	Optimization Of Parameters – Occurrences In Literature	56
3.3	Modelling 'Finding Optimal Parameters Of A Dynamic Stall Method' As An Optimization Problem	56
3.4	Error Computation And Scalarizing	57
3.5	Svandberg's Method Of Moving Asymptotes	59
3.6	Alternative Choices For An Optimization Method	67
3.6.1	Brute Force Optimization/Grid Search	67
3.6.2	Random Search/Random Optimization	68
3.6.3	Evolution Strategies	69
3.6.4	Implementation Of An Evolution Strategy	71
3.7	Parallel Computing	72
4	Optimization Of Time Constants And Parameters In Beddoes-Leishman Dynamic Stall Models	78
4.1	Optimization Of One Parameter	78
4.1.1	Test Case 101	78
4.1.2	Test Case 102	79
4.1.3	Test Case 103	79
4.1.4	Test Case 104	79
4.1.5	Test Case 105	79
4.1.6	Test Case 106	80
4.1.7	Test Case 107	80
4.1.8	Test Case 108	80
4.2	Optimization Of Several Parameters – S809 Airfoil	81
4.2.1	Test Case 101	81
4.2.2	Test Case 102	81
4.2.3	Test Case 103	81
4.2.4	Test Case 104	82
4.2.5	Test Case 105	82
4.2.6	Test Case 106	83
4.2.7	Test Case 107	83
4.2.8	Test Case 108	83
4.3	Optimization Of Several Parameters – S825 Airfoil	84

CONTENTS

4.4	Optimization Of Several Parameters – LS(1)-0147MOD Airfoil . .	85
5	Conclusions	86
5.1	Summary And Conclusions	86
5.2	Publications	87
A	Appendix	88
A.1	Figures – Performance Of NREL Dynamic Stall Model	88
A.1.1	Test Case 101	88
A.1.2	Test Case 102	90
A.1.3	Test Case 103	91
A.1.4	Test Case 104	93
A.1.5	Test Case 105	94
A.1.6	Test Case 106	96
A.1.7	Test Case 107	97
A.1.8	Test Case 108	99
A.2	Figures – Sensitivity Of The Dynamic Stall Model	100
A.2.1	Parameter A_1	100
	Test Case 101	100
	Test Case 102	102
	Test Case 103	103
	Test Case 104	105
	Test Case 105	106
	Test Case 106	108
	Test Case 107	109
	Test Case 108	111
A.2.2	Parameter A_2	113
	Test Case 101	113
	Test Case 102	114
	Test Case 103	116
	Test Case 104	117
	Test Case 105	119
	Test Case 106	120
	Test Case 107	122
	Test Case 108	123
A.2.3	Parameter b_1	125
	Test Case 101	125
	Test Case 102	126
	Test Case 103	128
	Test Case 104	129
	Test Case 105	131
	Test Case 106	132
	Test Case 107	134
	Test Case 108	135

CONTENTS

A.2.4	Parameter b_2	137
	Test Case 101	137
	Test Case 102	138
	Test Case 103	140
	Test Case 104	141
	Test Case 105	143
	Test Case 106	144
	Test Case 107	146
	Test Case 108	147
A.2.5	Time Constant T_f	149
	Test Case 101	149
	Test Case 102	150
	Test Case 103	152
	Test Case 104	153
	Test Case 105	155
	Test Case 106	156
	Test Case 107	158
	Test Case 108	159
A.2.6	Time Constant T_p	161
	Test Case 101	161
	Test Case 102	162
	Test Case 103	164
	Test Case 104	165
	Test Case 105	167
	Test Case 106	168
	Test Case 107	170
	Test Case 108	171
A.2.7	Time Constant T_v	173
	Test Case 101	173
	Test Case 102	174
	Test Case 103	176
	Test Case 104	177
	Test Case 105	179
	Test Case 106	180
	Test Case 107	182
	Test Case 108	183
A.2.8	Time Constant T_{VL}	185
	Test Case 101	185
	Test Case 102	186
	Test Case 103	188
	Test Case 104	189
	Test Case 105	191
	Test Case 106	192
	Test Case 107	194

	Test Case 108	195
A.3	Figures – Optimization Of One Parameter – S809 Airfoil	197
A.3.1	Test Case 101	197
A.3.2	Test Case 102	198
A.3.3	Test Case 103	200
A.3.4	Test Case 104	201
A.3.5	Test Case 105	203
A.3.6	Test Case 106	204
A.3.7	Test Case 107	206
A.3.8	Test Case 108	207
A.4	Figures – Optimization Of Multiple Parameters – S809 Airfoil	209
A.4.1	Test Case 101	209
A.4.2	Test Case 102	210
A.4.3	Test Case 103	212
A.4.4	Test Case 104	213
A.4.5	Test Case 105	215
A.4.6	Test Case 106	216
A.4.7	Test Case 107	218
A.4.8	Test Case 108	219
A.5	Figures – Optimization Of Multiple Parameters – S825 Airfoil	221
A.5.1	Test Case 101	221
A.5.2	Test Case 102	222
A.5.3	Test Case 103	224
A.5.4	Test Case 104	225
A.5.5	Test Case 105	227
A.5.6	Test Case 106	228
A.5.7	Test Case 107	230
A.5.8	Test Case 108	231
A.6	Figures – Optimization Of Multiple Parameters – LS(1)-417MOD Airfoil	233
A.6.1	Test Case 101	233
A.6.2	Test Case 102	234
A.6.3	Test Case 103	236
A.6.4	Test Case 104	237
A.6.5	Test Case 105	239
A.6.6	Test Case 106	240
A.6.7	Test Case 107	242
A.6.8	Test Case 108	243

LIST OF FIGURES

List of Figures

1.1	general schematic of lift curve	23
2.1	Forces in wind- and blade-based coordinate system	30
2.2	Separated flow over flat plate according to Kirchhoff theory . .	37
2.3	Separation point location f_N and its delayed counterparts f'_N and f''_N	38
2.4	Kinematics of test cases	45
2.5	Lift force over angle of attack α	46
2.6	Lift force over time t	47
3.1	Coupling of optimization algorithm and dynamic stall model .	57
3.2	Error over parameter T_{f0} for test case <i>TC106</i>	65
3.3	Error over parameter T_{f0} for test case <i>TC108</i>	66
3.4	Run-time	73
3.5	Speed-up	74
A.1	lift coefficient for test case 101	88
A.2	drag coefficient for test case 101	89
A.3	pitching moment coefficient for test case 101	89
A.4	lift coefficient for test case 102	90
A.5	drag coefficient for test case 102	90
A.6	pitching moment coefficient for test case 102	91
A.7	lift coefficient for test case 103	91
A.8	drag coefficient for test case 103	92
A.9	pitching moment coefficient for test case 103	92
A.10	lift coefficient for test case 104	93
A.11	drag coefficient for test case 104	93
A.12	pitching moment coefficient for test case 104	94
A.13	lift coefficient for test case 105	94
A.14	drag coefficient for test case 105	95
A.15	pitching moment coefficient for test case 105	95
A.16	lift coefficient for test case 106	96
A.17	drag coefficient for test case 106	96
A.18	pitching moment coefficient for test case 106	97
A.19	lift coefficient for test case 107	97

LIST OF FIGURES

A.20	drag coefficient for test case 107	98
A.21	pitching moment coefficient for test case 107	98
A.22	lift coefficient for test case 108	99
A.23	drag coefficient for test case 108	99
A.24	pitching moment coefficient for test case 108	100
A.25	lift coefficient for test case 101	100
A.26	drag coefficient for test case 101	101
A.27	pitching moment coefficient for test case 101	101
A.28	lift coefficient for test case 102	102
A.29	drag coefficient for test case 102	102
A.30	pitching moment coefficient for test case 102	103
A.31	lift coefficient for test case 103	103
A.32	drag coefficient for test case 103	104
A.33	pitching moment coefficient for test case 103	104
A.34	lift coefficient for test case 104	105
A.35	drag coefficient for test case 104	105
A.36	pitching moment coefficient for test case 104	106
A.37	lift coefficient for test case 105	106
A.38	drag coefficient for test case 105	107
A.39	pitching moment coefficient for test case 105	107
A.40	lift coefficient for test case 106	108
A.41	drag coefficient for test case 106	108
A.42	pitching moment coefficient for test case 106	109
A.43	lift coefficient for test case 107	109
A.44	drag coefficient for test case 107	110
A.45	pitching moment coefficient for test case 107	110
A.46	lift coefficient for test case 108	111
A.47	drag coefficient for test case 108	111
A.48	pitching moment coefficient for test case 108	112
A.49	lift coefficient for test case 101	113
A.50	drag coefficient for test case 101	113
A.51	pitching moment coefficient for test case 101	114
A.52	lift coefficient for test case 102	114
A.53	drag coefficient for test case 102	115
A.54	pitching moment coefficient for test case 102	115
A.55	lift coefficient for test case 103	116
A.56	drag coefficient for test case 103	116
A.57	pitching moment coefficient for test case 103	117
A.58	lift coefficient for test case 104	117
A.59	drag coefficient for test case 104	118
A.60	pitching moment coefficient for test case 104	118
A.61	lift coefficient for test case 105	119
A.62	drag coefficient for test case 105	119
A.63	pitching moment coefficient for test case 105	120

LIST OF FIGURES

A.64	lift coefficient for test case 106	120
A.65	drag coefficient for test case 106	121
A.66	pitching moment coefficient for test case 106	121
A.67	lift coefficient for test case 107	122
A.68	drag coefficient for test case 107	122
A.69	pitching moment coefficient for test case 107	123
A.70	lift coefficient for test case 108	123
A.71	drag coefficient for test case 108	124
A.72	pitching moment coefficient for test case 108	124
A.73	lift coefficient for test case 101	125
A.74	drag coefficient for test case 101	125
A.75	pitching moment coefficient for test case 101	126
A.76	lift coefficient for test case 102	126
A.77	drag coefficient for test case 102	127
A.78	pitching moment coefficient for test case 102	127
A.79	lift coefficient for test case 103	128
A.80	drag coefficient for test case 103	128
A.81	pitching moment coefficient for test case 103	129
A.82	lift coefficient for test case 104	129
A.83	drag coefficient for test case 104	130
A.84	pitching moment coefficient for test case 104	130
A.85	lift coefficient for test case 105	131
A.86	drag coefficient for test case 105	131
A.87	pitching moment coefficient for test case 105	132
A.88	lift coefficient for test case 106	132
A.89	drag coefficient for test case 106	133
A.90	pitching moment coefficient for test case 106	133
A.91	lift coefficient for test case 107	134
A.92	drag coefficient for test case 107	134
A.93	pitching moment coefficient for test case 107	135
A.94	lift coefficient for test case 108	135
A.95	drag coefficient for test case 108	136
A.96	pitching moment coefficient for test case 108	136
A.97	lift coefficient for test case 101	137
A.98	drag coefficient for test case 101	137
A.99	pitching moment coefficient for test case 101	138
A.100	lift coefficient for test case 102	138
A.101	drag coefficient for test case 102	139
A.102	pitching moment coefficient for test case 102	139
A.103	lift coefficient for test case 103	140
A.104	drag coefficient for test case 103	140
A.105	pitching moment coefficient for test case 103	141
A.106	lift coefficient for test case 104	141
A.107	drag coefficient for test case 104	142

LIST OF FIGURES

A.108	pitching moment coefficient for test case 104	142
A.109	lift coefficient for test case 105	143
A.110	drag coefficient for test case 105	143
A.111	pitching moment coefficient for test case 105	144
A.112	lift coefficient for test case 106	144
A.113	drag coefficient for test case 106	145
A.114	pitching moment coefficient for test case 106	145
A.115	lift coefficient for test case 107	146
A.116	drag coefficient for test case 107	146
A.117	pitching moment coefficient for test case 107	147
A.118	lift coefficient for test case 108	147
A.119	drag coefficient for test case 108	148
A.120	pitching moment coefficient for test case 108	148
A.121	lift coefficient for test case 101	149
A.122	drag coefficient for test case 101	149
A.123	pitching moment coefficient for test case 101	150
A.124	lift coefficient for test case 102	150
A.125	drag coefficient for test case 102	151
A.126	pitching moment coefficient for test case 102	151
A.127	lift coefficient for test case 103	152
A.128	drag coefficient for test case 103	152
A.129	pitching moment coefficient for test case 103	153
A.130	lift coefficient for test case 104	153
A.131	drag coefficient for test case 104	154
A.132	pitching moment coefficient for test case 104	154
A.133	lift coefficient for test case 105	155
A.134	drag coefficient for test case 105	155
A.135	pitching moment coefficient for test case 105	156
A.136	lift coefficient for test case 106	156
A.137	drag coefficient for test case 106	157
A.138	pitching moment coefficient for test case 106	157
A.139	lift coefficient for test case 107	158
A.140	drag coefficient for test case 107	158
A.141	pitching moment coefficient for test case 107	159
A.142	lift coefficient for test case 108	159
A.143	drag coefficient for test case 108	160
A.144	pitching moment coefficient for test case 108	160
A.145	lift coefficient for test case 101	161
A.146	drag coefficient for test case 101	161
A.147	pitching moment coefficient for test case 101	162
A.148	lift coefficient for test case 102	162
A.149	drag coefficient for test case 102	163
A.150	pitching moment coefficient for test case 102	163
A.151	lift coefficient for test case 103	164

LIST OF FIGURES

A.152	drag coefficient for test case 103	164
A.153	pitching moment coefficient for test case 103	165
A.154	lift coefficient for test case 104	165
A.155	drag coefficient for test case 104	166
A.156	pitching moment coefficient for test case 104	166
A.157	lift coefficient for test case 105	167
A.158	drag coefficient for test case 105	167
A.159	pitching moment coefficient for test case 105	168
A.160	lift coefficient for test case 106	168
A.161	drag coefficient for test case 106	169
A.162	pitching moment coefficient for test case 106	169
A.163	lift coefficient for test case 107	170
A.164	drag coefficient for test case 107	170
A.165	pitching moment coefficient for test case 107	171
A.166	lift coefficient for test case 108	171
A.167	drag coefficient for test case 108	172
A.168	pitching moment coefficient for test case 108	172
A.169	lift coefficient for test case 101	173
A.170	drag coefficient for test case 101	173
A.171	pitching moment coefficient for test case 101	174
A.172	lift coefficient for test case 102	174
A.173	drag coefficient for test case 102	175
A.174	pitching moment coefficient for test case 102	175
A.175	lift coefficient for test case 103	176
A.176	drag coefficient for test case 103	176
A.177	pitching moment coefficient for test case 103	177
A.178	lift coefficient for test case 104	177
A.179	drag coefficient for test case 104	178
A.180	pitching moment coefficient for test case 104	178
A.181	lift coefficient for test case 105	179
A.182	drag coefficient for test case 105	179
A.183	pitching moment coefficient for test case 105	180
A.184	lift coefficient for test case 106	180
A.185	drag coefficient for test case 106	181
A.186	pitching moment coefficient for test case 106	181
A.187	lift coefficient for test case 107	182
A.188	drag coefficient for test case 107	182
A.189	pitching moment coefficient for test case 107	183
A.190	lift coefficient for test case 108	183
A.191	drag coefficient for test case 108	184
A.192	pitching moment coefficient for test case 108	184
A.193	lift coefficient for test case 101	185
A.194	drag coefficient for test case 101	185
A.195	pitching moment coefficient for test case 101	186

LIST OF FIGURES

A.196	lift coefficient for test case 102	186
A.197	drag coefficient for test case 102	187
A.198	pitching moment coefficient for test case 102	187
A.199	lift coefficient for test case 103	188
A.200	drag coefficient for test case 103	188
A.201	pitching moment coefficient for test case 103	189
A.202	lift coefficient for test case 104	189
A.203	drag coefficient for test case 104	190
A.204	pitching moment coefficient for test case 104	190
A.205	lift coefficient for test case 105	191
A.206	drag coefficient for test case 105	191
A.207	pitching moment coefficient for test case 105	192
A.208	lift coefficient for test case 106	192
A.209	drag coefficient for test case 106	193
A.210	pitching moment coefficient for test case 106	193
A.211	lift coefficient for test case 107	194
A.212	drag coefficient for test case 107	194
A.213	pitching moment coefficient for test case 107	195
A.214	lift coefficient for test case 108	195
A.215	drag coefficient for test case 108	196
A.216	pitching moment coefficient for test case 108	196
A.217	lift coefficient for test case 101	197
A.218	drag coefficient for test case 101	197
A.219	pitching moment coefficient for test case 101	198
A.220	lift coefficient for test case 102	198
A.221	drag coefficient for test case 102	199
A.222	pitching moment coefficient for test case 102	199
A.223	lift coefficient for test case 103	200
A.224	drag coefficient for test case 103	200
A.225	pitching moment coefficient for test case 103	201
A.226	lift coefficient for test case 104	201
A.227	drag coefficient for test case 104	202
A.228	pitching moment coefficient for test case 104	202
A.229	lift coefficient for test case 105	203
A.230	drag coefficient for test case 105	203
A.231	pitching moment coefficient for test case 105	204
A.232	lift coefficient for test case 106	204
A.233	drag coefficient for test case 106	205
A.234	pitching moment coefficient for test case 106	205
A.235	lift coefficient for test case 107	206
A.236	drag coefficient for test case 107	206
A.237	pitching moment coefficient for test case 107	207
A.238	lift coefficient for test case 108	207
A.239	drag coefficient for test case 108	208

LIST OF FIGURES

A.240	pitching moment coefficient for test case 108	208
A.241	lift coefficient for test case 101	209
A.242	drag coefficient for test case 101	209
A.243	pitching moment coefficient for test case 101	210
A.244	lift coefficient for test case 102	210
A.245	drag coefficient for test case 102	211
A.246	pitching moment coefficient for test case 102	211
A.247	lift coefficient for test case 103	212
A.248	drag coefficient for test case 103	212
A.249	pitching moment coefficient for test case 103	213
A.250	lift coefficient for test case 104	213
A.251	drag coefficient for test case 104	214
A.252	pitching moment coefficient for test case 104	214
A.253	lift coefficient for test case 105	215
A.254	drag coefficient for test case 105	215
A.255	pitching moment coefficient for test case 105	216
A.256	lift coefficient for test case 106	216
A.257	drag coefficient for test case 106	217
A.258	pitching moment coefficient for test case 106	217
A.259	lift coefficient for test case 107	218
A.260	drag coefficient for test case 107	218
A.261	pitching moment coefficient for test case 107	219
A.262	lift coefficient for test case 108	219
A.263	drag coefficient for test case 108	220
A.264	pitching moment coefficient for test case 108	220
A.265	lift coefficient for test case 101	221
A.266	drag coefficient for test case 101	221
A.267	pitching moment coefficient for test case 101	222
A.268	lift coefficient for test case 102	222
A.269	drag coefficient for test case 102	223
A.270	pitching moment coefficient for test case 102	223
A.271	lift coefficient for test case 103	224
A.272	drag coefficient for test case 103	224
A.273	pitching moment coefficient for test case 103	225
A.274	lift coefficient for test case 104	225
A.275	drag coefficient for test case 104	226
A.276	pitching moment coefficient for test case 104	226
A.277	lift coefficient for test case 105	227
A.278	drag coefficient for test case 105	227
A.279	pitching moment coefficient for test case 105	228
A.280	lift coefficient for test case 106	228
A.281	drag coefficient for test case 106	229
A.282	pitching moment coefficient for test case 106	229
A.283	lift coefficient for test case 107	230

A.284	drag coefficient for test case 107	230
A.285	pitching moment coefficient for test case 107	231
A.286	lift coefficient for test case 108	231
A.287	drag coefficient for test case 108	232
A.288	pitching moment coefficient for test case 108	232
A.289	lift coefficient for test case 101	233
A.290	drag coefficient for test case 101	233
A.291	pitching moment coefficient for test case 101	234
A.292	lift coefficient for test case 102	234
A.293	drag coefficient for test case 102	235
A.294	pitching moment coefficient for test case 102	235
A.295	lift coefficient for test case 103	236
A.296	drag coefficient for test case 103	236
A.297	pitching moment coefficient for test case 103	237
A.298	lift coefficient for test case 104	237
A.299	drag coefficient for test case 104	238
A.300	pitching moment coefficient for test case 104	238
A.301	lift coefficient for test case 105	239
A.302	drag coefficient for test case 105	239
A.303	pitching moment coefficient for test case 105	240
A.304	lift coefficient for test case 106	240
A.305	drag coefficient for test case 106	241
A.306	pitching moment coefficient for test case 106	241
A.307	lift coefficient for test case 107	242
A.308	drag coefficient for test case 107	242
A.309	pitching moment coefficient for test case 107	243
A.310	lift coefficient for test case 108	243
A.311	drag coefficient for test case 108	244
A.312	pitching moment coefficient for test case 108	244

List of Tables

2.1	Overview of kinematics of dynamic test cases	45
2.2	Default values of parameters and time constants for S809 airfoil	49
3.1	Master slave framework	76
3.2	Overview of speed-up properties	77
4.1	Overview of design variables (S809)	81

LIST OF TABLES

4.2	Overview of design variables (S825)	84
4.3	Overview of design variables (LS(1)-0417MOD)	85

Nomenclature

Abbreviations

<i>AeroDyn</i>	wind turbine aerodynamics analysis software by the NREL (page 28)
<i>BEM</i>	blade element momentum (page 28)
<i>CENER</i>	National Renewable Energy Centre (Spain) (page 26)
<i>CPU</i>	central processing unit (page 72)
<i>DYNSTALL</i>	dynamic stall model by FFA (page 26)
<i>DYSTOOL</i>	dynamic stall model developed by CENER (page 26)
<i>ES</i>	Evolution Strategy (page 69)
<i>FAST</i>	wind turbine multiphysics analysis software by NREL (page 28)
<i>FFA</i>	Flygtekniska Försöksanstalten (National Aeronautical Research Institute) of Sweden (page 26)
<i>MMA</i>	Method of Moving Averages (page 26)
<i>MPI</i>	message passing interface (page 72)
<i>NASA</i>	National Aeronautics and Space Administration (page 85)
<i>NREL</i>	National Renewable Energy Laboratory (page 28)
<i>NTU</i>	Nanyang Technological University (page 72)
<i>UAE</i>	Unsteady Aerodynamics Experiment (page 56)

Greek Symbols

α	angle of attack (page 29)
α'_f	delayed effective angle of attack (page 41)
α_0	zero-lift angle of attack (page 32)

LIST OF TABLES

α_{amp}	amplitude angle in sinusoidal pitch motion (page 44)
α_e	effective angle of incidence (page 32)
α_f	effective angle of attack (page 39)
α_{mean}	mean angle in sinusoidal pitch motion (page 44)
β	Prandtl-Glauert compressibility correction factor $\beta = \sqrt{1 - M^2}$ (page 31)
κ	non-dimensional frequency $\left(\kappa = \frac{c}{2U_{res}} \omega \right)$ (page 44)
λ	arbitrary parameter (page 63)
ω	angular frequency (page 44)
ϕ	indicial function (page 30)
$\phi_{M,\alpha}^c$	circulatory indicial response function of pitching moment due to change of α (page 34)
$\phi_{M,q}^c$	circulatory indicial response function of pitching moment due to change in pitch rate q (page 34)
$\phi_{N,\alpha}^c$	circulatory indicial response function of normal force due to change of angle of attack α (page 31)
$\phi_{N,q}^c$	circulatory indicial response function of normal force due to change in pitch rate q (page 31)
$\phi_{M,\alpha}^{nc}$	non-circulatory indicial response function of pitching moment due to change of α (page 34)
$\phi_{M,q}^{nc}$	non-circulatory indicial response function of pitching moment due to change in pitch rate q (page 34)
$\phi_{N,\alpha}^{nc}$	non-circulatory indicial response function of normal force due to change of angle of attack α (page 31)
$\phi_{N,q}^{nc}$	non-circulatory indicial response function of normal force due to change in pitch rate q (page 31)
τ_V	time variable to track leading edge vortex travelling over chord (page 42)
ϵ_D	error in drag (page 58)
ϵ_D	error in drag (page 58)

LIST OF TABLES

ϵ_L	error in lift (page 58)
ϵ_M	error in pitching moment (page 58)
ϵ_M	error in pitching moment (page 58)
ϵ_{total}	total error (page 58)

Latin Symbols

a	speed of sound (page 32)
A_1	empirical constant ($A_1 = 0.3$) (page 31)
A_2	empirical constant ($A_2 = 0.7$) (page 31)
A_3	empirical constant (page 34)
A_4	empirical constant (page 34)
A_5	empirical constant ($A_5 = 1.00$) (page 34)
a_{cd}	correction factor for an assumed vortex drag in the DYNSTALL model (page 62)
b_1	empirical constant ($b_1 = 0.14$) (page 31)
b_2	empirical constant ($b_2 = 0.53$) (page 31)
b_3	empirical constant (page 34)
b_4	empirical constant (page 34)
b_5	empirical constant ($b_5 = 5.00$) (page 34)
C	arbitrary, convex function (page 64)
c	chord (page 29)
c'_N	effective normal force coefficient (page 39)
$c_{N,\alpha q}^c$	circulatory normal force response due to change in angle of attack and pitch rate (page 32)
$c_{N,\alpha}^{nc}$	non-circulatory normal force response due to change in angle of attack (page 33)
$c_{N,q}^{nc}$	non-circulatory normal force response due to change in pitch rate (page 34)
c_C^v	vortex-induced component of chordwise force coefficient (page 43)

LIST OF TABLES

c_M^v	vortex-induced component pitching moment (page 44)
c_N^v	vortex-induced component of normal force coefficient (page 43)
$c_{C,pot}$	chordwise force under attached flow (page 34)
c_C	chord force coefficient (page 29)
c_{D_0}	drag coefficient at zero lift (page 29)
c_D	drag coefficient (page 29)
$c_{L,\alpha}$	slope of lift curve (page 31)
c_L	lift coefficient (page 29)
$c_{M,pot}$	pitching moment under attached flow (page 34)
c_{M_0}	zero lift pitching moment coefficient (page 34)
$c_{N,pot}$	normal force under attached flow (page 34)
c_{N1}	critical normal force for leading edge separation to occur (page 42)
c_{N2}	critical normal force for leading edge separation to occur (at negative angles of attack) (page 42)
c_N	normal force coefficient (page 29)
c_V	normal force contribution due to accumulated leading edge vorticity (page 43)
D	dual (optimization) problem (page 61)
d	coefficient of artificial variable in Method of Moving Averages (page 63)
D_{α_f}	deficiency function for α'_f computation (page 41)
D_{f_C}	deficiency function for f''_C computation (page 40)
D_{f_N}	deficiency function for f''_N computation (page 39)
D_p	deficiency function for c'_N computation (page 39)
$e_{L,i}$	error in drag at time step i (page 58)
f	separation point location (page 37)
f''_C	delayed effective separation point location for the chordwise force treatment (page 40)

LIST OF TABLES

f_N''	delayed effective separation point location for the normal force treatment (page 39)
f_C'	effective separation point location for the chordwise force treatment (page 40)
f_N'	effective separation point location for the normal force treatment (page 39)
f_C	separation point location for the chordwise force treatment (page 38)
f_N	separation point location for the normal force treatment (page 37)
f_p	fraction of execution time that can be parallelized (page 74)
g	objective function (page 55)
$\hat{g}$	right-hand side term of constraints (page 59)
h	(inequality) constraint function (page 55)
i	(equality) constraint function (page 55)
K_q'''	deficiency function for $c_{M,q}^c$ computation (page 36)
K_q''	deficiency function for $c_{M,q}^{nc}$ computation (page 36)
K_α'	deficiency function for $c_{N,\alpha}^{nc}$ computation (page 33)
K_q'	deficiency function for $c_{N,q}^{nc}$ computation (page 34)
$k_{M,q}$	constant (page 36)
L	quantity in method of moving averages (lower moving asymptote) (page 60)
M	Mach number (page 31)
n	number of CPUs (page 74)
N_{elite}	number of best sample solutions to retain without recombination (page 71)
N_{gen}	number of generations (page 71)
N_{new}	number of freshly generated sample solutions each generation (page 71)
$N_{samples}$	number of sample solutions per generation (page 71)
P	primary (optimization) problem (page 59)

LIST OF TABLES

p	quantity in method of moving averages (page 60)
$\tilde{P}^{(k)}$	modified sub-problem of k -th iteration (page 63)
$P^{(k)}$	sub-problem of k -th iteration (page 61)
q	(non-dimensionalized) pitch rate $\left(q = \frac{c}{U_{res}}\right) \dot{\alpha}$ (page 31)
q	quantity in method of moving averages (page 60)
s	non-dimensional time $\left(s = t \frac{2U_{res}}{c}\right)$ (page 31)
s	speed-up through parallelization (page 74)
$\mathbb{S}$	convex set in $\mathbb{R}^n$ (page 63)
T	period of one cycle (page 48)
T_{f0}	initial value of T_f (page 39)
T_f	empirical time constant (page 39)
T_f	time constant representing delayed separation point response (page 62)
$T_{M,\alpha}$	empirical time constant (page 35)
$T_{M,q}$	empirical time constant (page 35)
$T_{N,\alpha}$	empirical time constant ($T_{N,\alpha} \rightarrow 1.0$ for $M \rightarrow 0$) (page 31)
$T_{N,q}$	empirical time constant ($T_{N,q} \rightarrow 1.0$ for $M \rightarrow 0$) (page 31)
T_p	empirical time constant (page 39)
T_p	time constant representing boundary layer leading edge pressure gradient delay (page 62)
T_{V0}	initial value of T_V (page 43)
T_{VL}	time constant (denotes time for leading edge vortex to travel along chord) (page 42)
T_V	empirical time constant (page 43)
T_V	time constant in the vortex lift expression (page 62)
U	quantity in method of moving averages (upper moving asymptote) (page 60)
U_{blade}	(rotational) velocity of the blade (page 30)

LIST OF TABLES

U_{res}	resulting velocity (page 29)
U_{res}	resulting velocity (page 30)
U_{wind}	wind velocity (page 30)
W	objective function (of the dual (optimization) problem) (page 61)
w_D	weighting factor for drag error term (page 59)
w_L	weighting factor for lift error term (page 59)
w_M	weighting factor for pitching moment error term (page 59)
x	design variable (page 55)
X_1	deficiency function for α_e computation (page 33)
X_2	deficiency function for α_e computation (page 33)
x_{ac}	(non-dimensionalized) location of aerodynamic center along chord (page 34)
x_{cp}	(non-dimensionalized) location of center of pressure along chord (page 44)
x_{max}	upper bound for design variable (page 59)
x_{min}	lower bound for design variable (page 59)
y	design variable (of the dual (optimization) problem) (page 61)
z	artificial variable in Method of Moving Averages (page 63)

Subscripts/Superscripts

$\underline{\quad}$ (<i>underline</i>)	vector quantity (page 55)
c	circulatory (page 31)
exp	quantity provided by experiment (page 58)
nc	non-circulatory (page 31)
sim	quantity produced by simulation (dynamic stall model) (page 58)

Chapter 1

Background

1.1 Dynamic Stall

Stall is an aerodynamic phenomenon that describes the loss of lift due to flow separation.

Classical thin-airfoil theory relates the lift force – or rather the lift coefficient c_L – of an airfoil to its angle of attack α via the linear relationship:

$$c_L = 2\pi\alpha. \quad (1.1)$$

When increasing the angle of attack, stall – a sudden loss of lift – can be observed when reaching a certain angle, the *critical angle of attack*. This is due to flow separation occurring across the suction side of the airfoil. Beyond that critical angle of attack, no more lift increase is possible. A typical lift curve might look like the full line in Figure 1.1. However, there are types of airfoils for which stall does not occur suddenly, but rather gradually. The loss of lift is usually accompanied by increased drag due to the recirculating flow caused by the separation and by a sharp reduction of pitching moment due to the loss of suction pressure on the upper surface.

Dynamic stall occurs when the airfoil is subjected to a change in the angle of attack (e.g. due to pitching or plunging motion). It describes the phenomenon that the flow remains attached even though the critical angle of attack might be exceeded. Taking the example of an airfoil performing a pitch-up motion, the lift curve will look like the lift curve of the static case as long as the angle of attack remains below the critical value of α . Approaching this critical angle of attack α_{crit} , the onset of separation is delayed to an angle larger than α_{crit} , which is depicted in the dotted

1.1. DYNAMIC STALL

Figure 1.1: general schematic of lift curve

curve in Figure 1.1.

Dynamic stall is associated with shedding of vorticity from the leading edge which can – depending on the circumstances – form one or several vortices travelling over the suction side of the airfoil. While progressing over the airfoil, lift is enhanced – often in an unsteady, oscillating fashion. Once the travelling vortex passes the trailing edge, sudden and heavy loss of lift occurs and the airfoil returns to regular stalled state. Together with the effect on lift, the leading edge vortex also affects the pitching moment and an amplification of the (nose-down pitching) moment reduction can be observed. This is due to the lift increase, but also due to the movement of the location of the centre of pressure.

The delay of stall can be advantageous, as it means higher performance in terms of lift (production). However, this is only temporary, since the dynamic stall ends soon after either the motion recedes or the critical angle of dynamic stall is reached and the flow also becomes fully separated.

As another example, consider an airfoil undergoing periodic pitch-up and pitch-down motion: it will not only experience delayed stall during the pitch-up motion, but also a delay in the reattachment from separated flow when pitching back down. As a result of this, the lift curve will show a hysteresis pattern with reduced aerodynamic damping and the periodic excitations might lead to so-called *stall flutter* or other detrimental phenomena involving vibrations, high stresses, etc. As an example, the rapid nose-down pitching moment, which occurs when the vortex shed from the leading edge travels over the blade surface, induces torsional loads on the

1.2. APPLICATION OF DYNAMIC STALL

airfoil.

In order to deal with the beneficial as well as the detrimental effects of dynamic stall, careful considerations during design of airfoils and lifting surfaces are required, providing reason for why models and tools for predicting dynamic stall and the resulting loads are necessary. This is significant for the performance, but often more so for issues of long-term loading, fatigue and longevity.

1.2 Application Of Dynamic Stall

Fields in aerodynamics in which dynamic stall becomes relevant involve aerodynamics of rotating blades, most notably aerodynamics of helicopters, which usually involves more complex settings because of the inherent three-dimensionality of the flow due to the blade rotation. Blade element momentum (BEM) methods are a common tool for load analysis of helicopter rotors. In order to make up for their inability to deal with unsteady loads, dynamic stall models have been integrated into BEM methods.

Besides that, wind turbines are another field of application. What is different to helicopter aerodynamics is the low Mach number regime rendering effects of incompressibility negligible. There are stall-controlled turbines with fixed blades whose performance is governed by stalling. With unsteady inflow velocity due to gusts, sheared or yawed winds – a change in the incoming wind speed comes with a change in the angle of attack felt by the turbine blade – this gives different operating conditions requiring accurate load prediction. The other type, so called pitch-controlled wind turbines control their power output by pitching the blades continuously depending on speed and direction of the incoming wind. The pitching of blades marks another scenario for which knowledge of dynamic loads is desired.

Even more examples, in which dynamic stall models might play a role, involve aerodynamics of flapping wings, e.g. simulating the flight of birds and insects, other propeller flow applications, etc.

A dynamic stall model can relate the incoming flow in terms of flow speed and angle of attack to the resulting forces. Thus, a dynamic stall model by itself is not of much use, only in connection with other components providing input to a dynamic stall model, a useful tool for engineering and research purposes is generated. Use cases for such an implementation are integrating a dynamic stall model into a BEM program for creating a software to analyze helicopter blade loads or into a vortex

1.3. MODELLING OF DYNAMIC STALL

wake method for prediction the performance of a wind turbine.

1.3 Modelling Of Dynamic Stall

Having explained the necessity for insight into the effects of dynamic stall, this section describes some of the modelling approaches from history.

Gross and Harris [1] (and later continued by Gormont[2]) used Theodorsen's theory to derive an "effective" angle of attack representing the delay in α for an airfoil undergoing pitching motion. Forces could then be determined from static forces stored look-up tables. This method could give a reasonable estimate for the delay of separation, but it could not help with prediction of stall onset itself.

Beddoes is considered one of the "fathers" of dynamic stall modelling. During his time at Westland Helicopters, he developed a model together with Leishman ([3], [4]) based on so-called *indicial functions* describing the aerodynamic response to a disturbance which, in this sense, can be an increase in the angle of attack from one time step to the next. By finding suitable approximations to these indicial functions, it becomes possible to express forces acting on a blade for arbitrary pitch and plunge motions under inviscid flow. Separated flow is treated using Kirchhoff-Helmholtz theory relating the lift to the degree of separation. A third component treats the shedding and travelling of the leading edge vortex along the blade chord. Beddoes also introduced a criterion for stall onset based on an equivalent leading edge pressure. The model of Leishman and Beddoes has seen widespread use and Leishman continued the work in this field during his tenure at the University of Maryland. In particular, in 2006, Gupta [5], a PhD student under Leishman, presented his application of the Leishman-Beddoes model to the low-speed case of a wind turbine.

In 1982, Viterna and Jonetzke [6] developed a stall model based on flat-plate theory which was applied extensively in wind turbine research. More stall delay models for wind turbines have been proposed by Du and Selig [7] in 1998 and Corrigan and Schillings [8] in 1994. In 1995, Pierce and Hansen [9] applied the Leishman-Beddoes model to predict the blade loads with dynamic stall.

Even though computer hardware has progressed tremendously over the last decades, Computational Fluid Dynamics (CFD) are neither widespread nor recognized in the research of dynamic stall. This is due to unresolved issues of turbulence modelling, boundary layer transition and most notably separation prediction.

To summarize this section, dynamic stall models are either completely empirical,

1.4. MOTIVATION FOR THIS WORK

i.e. based on observations from wind tunnel experiments, or semi-empirical models derived from classical thin airfoil theory. No matter what degree of empiricism, they often involve constants or coefficients which are tediously derived and work only for particular airfoils or under certain conditions (Mach number, Reynolds number, etc.).

1.4 Motivation For This Work

Beddoes-Leishman dynamic stall models rely on various time constants and parameters to be specified by the user prior to execution. These parameters represent airfoil-specific properties and behaviour. Some of them are characteristic properties determined from static wind tunnel tests, such as zero-lift angle of attack or the slope of the lift curve. Others, e.g. time constants quantifying certain delays, are purely empirical and can neither be analytically derived nor determined based on sound scientific method. Instead, suitable values for such parameters and time constants in a dynamic stall model are found by *trial & error*. By playing around with the parameters and comparing to experimental data, the user will eventually arrive at a proper set of parameters. However, the question remains whether this will represent the best possible, the *optimal* set or not. Furthermore, this crude way of optimization requires the constant focus and attention of the user, so it is questionable whether this is adequate for an industrial setting. Hence, there is a clear need for methods to determine such suitable sets of parameters. Besides that, an additional benefit would be generated from executing such a method without constant supervision or if it would even run completely automatized. Optimization methods feature those kind of qualities and hence, can satisfy the mentioned requirements.

As of today, there are only two documented instances where proper optimization methods instead of 'trial & error' have been applied to find suitable values for time constants or other empirical parameters in dynamic stall models of Beddoes-Leishman type: Mert [10] applied the so-called method of moving asymptotes (MMA), an optimization algorithm developed by Svanberg [11], to find optimal parameters T_f and T_v for DYNSTALL [12], a dynamic stall model developed at the Swedish Aeronautical Research Institute (FFA). Gonzalez and Munduate of the Spanish National Renewable Energy Centre (CENER) have developed a Beddoes-Leishman type dynamic stall model named DYSTOOL [13], for which they used a genetic algorithm to determine necessary model parameters [14].

1.4. MOTIVATION FOR THIS WORK

An optimization method for finding not only two parameters like Mert [10], but all, would mark a great improvement over the current state. Also, making the software available to the public via an *open-source* approach will benefit the wind energy community from layman to expert by providing a tool for calibrating a dynamic stall model to a new airfoil geometry.

Chapter 2

Dynamic Stall Model

A dynamic stall model is employed in conjunction with a model for predicting aerodynamic loads on the blades. This could be a *Blade Element Momentum* (BEM) *model* or – as it is becoming more popular in the wind energy community – a *Dynamic Wake model*. These models essentially take the wind speed and blade kinematics as input and deliver integrated loads of lift, drag, pitching moment as output. They do, however, not take into account unsteady effects such as delayed separation and reattachment, for which a dedicated dynamic stall model is included. In terms of software architecture, it needs to be executed after the primary load prediction model and receives wind speed, blade kinematics and force history (from previous time step) as input, based on which it then computes loads. So, the process chain involves a BEM or dynamic wake model as the load prediction tool and the dynamic stall model as a postprocessor to the primary load prediction tool.

The dynamic stall model investigated in this work is AeroDyn [15], an open source wind turbine aerodynamics analysis software by the National Renewable Energy Laboratory (NREL).

2.1 The AeroDyn Dynamic Stall Model

AeroDyn is available as a stand-alone software (current version: 15) [15] or as a module of FAST (current version: 8) [16], the wind turbine multiphysics analysis tool by the NREL. In both implementations, it is an *open source* software. The dynamic stall model in AeroDyn is a Beddoes-Leishman-type model, named after its original developers Beddoes and Leishman [3, 4].

2.1. THE AERODYN DYNAMIC STALL MODEL

The Beddoes-Leishman-type models usually consist of several components or sub-models executed one after the other in a so-called *Kelvin chain*, a linear link-up in which the output of one component is taken as input for the next component. Even though they are executed sequentially, they are still physically coupled. As an example, while there is leading edge separation ongoing, reattachment will be delayed, although the angle of attack might have already reduced below the static separation angle. Such interaction of the phenomena is achieved by modification of several time constants associated with boundary layer response, vortex shedding, etc. This approach is easily implemented and avoids additional equations.

The principle idea of these submodels is to develop expressions for the forces normal and tangential to the airfoil chord as well as the pitching moment under different conditions of attached or separated flow. The different phenomena themselves are mathematically modelled as first-order systems consisting of differential equations. The time constants appearing in these differential equations are responsible for giving airfoil- or environmental-specific responses, i.e. they need to be chosen properly to model the behaviour of different airfoil shapes under different conditions, which is usually achieved via matching to experimental results. The designated components, or submodels, for attached flow, separated flow and vortex shedding will be described in the subsequent sections. The entire model is set up using non-dimensionalized quantities, so when speaking of forces and moment, the actual quantities are force and moment coefficients. The scaling applied is based on $c/2$ (half a chord length) in space and $2U_{res}/c$ (the time necessary to travel half a chord at a speed of U_{res}) in time.

The normal and tangential forces c_N and c_C are defined in a coordinate system attached to the airfoil, i.e. rotating with the pitching motion of the airfoil. They can be converted into lift and drag (c_L and c_D), the forces defined in a “wind-based” coordinate system, via a simple coordinate transformation with the angle of attack α :

$$c_L = c_N \cdot \cos(\alpha) + c_C \cdot \sin(\alpha) \quad (2.1)$$

and

$$c_D = c_N \cdot \sin(\alpha) - c_C \cdot \cos(\alpha) + c_{D0}. \quad (2.2)$$

Figure 2.1 shows this geometric relationship. Besides the forces (force coefficients, actually) in the two coordinate systems (wind-based and airfoil-based), it

2.1. THE AERODYN DYNAMIC STALL MODEL

also shows the incoming wind speed U_{wind} , which yields – together with the rotational velocity of the blade U_{blade} – the resulting wind speed U_{res} “felt” by the blade.

Figure 2.1: Forces in wind- and blade-based coordinate system

The dynamic stall model in AeroDyn is a pure two-dimensional model, i.e. two-dimensional in the plane spanned by coordinate systems of figure 2.1. Dynamic effects arising from three-dimensional conditions such as yawed flow, Coriolis or centrifugal forces are not included. However, some of these effects are accounted for by other, designated models implemented into AeroDyn.

2.1.1 Unsteady Attached Flow (Submodel)

As it was mentioned in Section 1.3, Leishman-Beddoes models use so-called *indicial aerodynamic functions* ϕ to describe the transient, aerodynamic response to a series of disturbances during attached flow. Disturbances in this context are changes in angle of attack or in pitch rate governed by a step function. Derived from linearized equations of potential flow, indicial functions consist of circulatory and non-circulatory contribution. The *circulatory* component builds up from zero asymptotically to a steady state value, it is supposed to represent the build-up of lift-generating circulation around the airfoil as the boundary layer develops or adjusts to the new flow field. The *non-circulatory* response (also called impulsive response), on the other hand, describes the initial, but rapidly decaying loading in response to an instantaneous disturbance, it is caused by waves of compression and

2.1. THE AERODYN DYNAMIC STALL MODEL

rarefaction on the opposing faces of the airfoil. In order to account for the decaying behaviour, the non-circulatory loading is modelled via decaying exponential functions. If the indicial response functions can be determined, then the load response to an arbitrary forcing can be computed according to the so-called Duhamel's superposition principle. A detailed derivation of the different indicial functions can be found in [17].

Normal Force

If these indicial aerodynamic functions are approximated properly, the normal force c_N for arbitrary changes in angle of attack α as well as in pitch rate q (non-dimensionalized pitch rate $q = \frac{c}{U_{res}} \dot{\alpha}$) can be computed in a numerical scheme. In the following equations, the superscripts c and nc shall stand for circulatory and non-circulatory component, respectively. The subscripts N, α and N, q represent normal force due to disturbance in α and pitch rate q , respectively, and accordingly subscripts C, α and C, q for the chordwise force as well as M, α and M, q for the pitching moment.

The normal force under attached flow (index *pot* for “potential”) can be computed by the following expression:

$$c_{N,pot} = \alpha \cdot \left(\frac{c_{L,\alpha}}{\beta} \cdot \phi_{N,\alpha}^c + \frac{4}{M} \cdot \phi_{N,\alpha}^{nc} \right) + q \cdot \left(\frac{c_{L,\alpha}}{2\beta} \cdot \phi_{N,q}^c + \frac{1}{M} \cdot \phi_{N,q}^{nc} \right). \quad (2.3)$$

In this equation, M denotes the Mach number and β the Prandtl-Glauert compressibility factor ($\beta = \sqrt{1 - M^2}$), which can be assumed as 1 for wind turbine applications, it should only have an impact in the regime of helicopter blades, where Mach numbers larger than 0.3 can be reached and compressibility effects play a role. $c_{L,\alpha}$ represents the slope of the (static) lift curve (i.e. c_L over α). The four indicial response functions $\phi_{N,\alpha}^c$ (circulatory indicial response function of normal force due to change of angle of attack α), $\phi_{N,\alpha}^{nc}$ (non-circulatory indicial response function of normal force due to change of angle of attack α), $\phi_{N,q}^c$ (circulatory indicial response function of normal force due to change in pitch rate q), $\phi_{N,q}^{nc}$ (non-circulatory indicial response function of normal force due to change in pitch rate q) are formulated – according to Leishman [18] – as follows:

$$\phi_{N,\alpha}^c = 1.0 - A_1 \cdot \exp(b_1 \cdot \beta^2 s) - A_2 \cdot \exp(b_2 \cdot \beta^2 s), \quad (2.4)$$

2.1. THE AERODYN DYNAMIC STALL MODEL

$$\phi_{N,\alpha}^{nc} = \exp\left(-\frac{c}{2U_{res}T_{N,\alpha}}s\right), \quad (2.5)$$

$$\phi_{N,q}^c = \phi_{N,\alpha}^c \quad (2.6)$$

and

$$\phi_{N,q}^{nc} = \exp\left(-\frac{c}{2U_{res}T_{N,q}}s\right). \quad (2.7)$$

In these equations, the time has been non-dimensionalized according to:

$$s = \frac{2U_{res}}{c}t. \quad (2.8)$$

The expressions of the $\phi_{N,\alpha}^c$ and $\phi_{N,q}^c$ involve four constants A_1 and A_2 , b_1 and b_2 , for which Beddoes [19] had derived values: $A_1 = 0.3$ and $A_2 = 0.7$, $b_1 = 0.14$ and $b_2 = 0.53$. He derived them by approximating indicial responses to several NACA airfoils (*NACA0006*, *NACA0012*, *NACA0012MOD*, *NACA64A010*) at different Mach numbers. Furthermore, there are two Mach number dependent time constants $T_{N,\alpha}$ and $T_{N,q}$ involved in the above equations, both approaching $\frac{c}{a}$ for $M \ll 1$, as is the case in this work, where a denotes the speed of sound. The exact expressions for $T_{N,\alpha}$ and $T_{N,q}$ read:

$$T_{N,\alpha} = \frac{c}{a(1-M) + \frac{1}{2}ac_{L,\alpha}M^2\beta(A_1b_1 + A_2b_2)} \quad (2.9)$$

and

$$T_{N,q} = \frac{c}{a(1-M) + ac_{L,\alpha}M^2\beta(A_1b_1 + A_2b_2)}. \quad (2.10)$$

The circulatory response of the normal force, i.e. the two terms involving $\phi_{N,\alpha}^c$ and $\phi_{N,q}^c$, can be expressed in terms of an *effective angle of incidence* α_e multiplied by the lift curve slope $c_{L,\alpha}$ via the so-called Duhamel integral. This allows to compute two of the four terms in Equation (2.3) conveniently:

$$\alpha \cdot \frac{c_{L,\alpha}}{\beta} \cdot \phi_{N,\alpha}^c + q \cdot \frac{c_{L,\alpha}}{2\beta} \cdot \phi_{N,q}^c = c_{N,\alpha q}^c = c_{L,\alpha} \cdot \alpha_e, \quad (2.11)$$

where α_e (at time step s_i) is computed as follows:

$$\alpha_e(s_i) = \alpha(s_i) - \alpha_0 - X_1(s_i) - X_2(s_i). \quad (2.12)$$

2.1. THE AERODYN DYNAMIC STALL MODEL

In this expression, α_0 is the zero-lift angle of attack, X_1 and X_2 are the recursive, so-called *deficiency functions* defined as:

$$X_1(s_i) = X_1(s_{i-1}) \cdot \exp(-b_1 \beta^2 \Delta s) + A_1 \cdot \exp(-b_1 \beta^2 \frac{\Delta s}{2}) \quad (2.13)$$

and

$$X_2(s_i) = X_2(s_{i-1}) \cdot \exp(-b_2 \beta^2 \Delta s) + A_2 \cdot \exp(-b_2 \beta^2 \frac{\Delta s}{2}). \quad (2.14)$$

They have the common structure of two exponential terms with the first representing the particular quantity's decaying behaviour (recognizable by the recursive appearance of the quantity from the previous time step) and the second term providing the contribution during the current time step. In a non-mathematical sense, they represent the deficiency in terms of angle of attack due to unsteady effects and the history of the loads due to the shed wake. Note that the notation $\Delta s = s_i - s_{i-1}$ for a time step size has been used. By applying Equation (2.12) to the current angle of attack $\alpha(s_i)$ and then using α_e for the computation of the circulatory loads, the lag characteristic for the unsteady effects can be achieved, representing a fundamental concept of the Leishman-Beddoes type dynamic stall models.

Such a comprehensive expression involving the lift curve slope and an effective angle of attack as for the circulatory response is not available for the non-circulatory terms, they need to be approximated independently. The non-circulatory response to a change in angle of attack is computed according to:

$$c_{N,\alpha}^{nc}(s_i) = \frac{4T_{N,\alpha}}{M} \left(\frac{\alpha(s_i) - \alpha(s_{i-1})}{\Delta s} - K'_\alpha(s_i) \right). \quad (2.15)$$

It is computed from a finite difference in α and a deficiency function K'_α that is supposed to account for time history effects of wave-like pressure disturbances.

$$K'_\alpha(s_i) = K'_\alpha(s_{i-1}) \cdot \exp\left(\frac{-\Delta s}{T_{N,\alpha}}\right) + \frac{\alpha(s_i) - 2\alpha(s_{i-1}) + \alpha(s_{i-2})}{\Delta s} \cdot \exp\left(\frac{-\Delta s}{2T_{N,\alpha}}\right). \quad (2.16)$$

2.1. THE AERODYN DYNAMIC STALL MODEL

The approximation of the non-circulatory response to a change in the pitch rate has a very similar structure:

$$c_{N,q}^{nc}(s_i) = -\frac{T_{N,q}}{M} \left(\frac{q(s_i) - q(s_{i-1})}{\Delta s} - K'_q(s_i) \right). \quad (2.17)$$

It is also computed from a finite difference and a deficiency function K'_q :

$$K'_q(s_i) = K'_q(s_{i-1}) \cdot \exp\left(\frac{-\Delta s}{T_{N,q}}\right) + \frac{q(s_i) - 2q(s_{i-1}) + q(s_{i-2}))}{\Delta s} \cdot \exp\left(\frac{-\Delta s}{2T_{N,q}}\right). \quad (2.18)$$

The normal force under unsteady, attached flow is then the sum of the circular response (2.11) and both non-circulatory terms (2.15) and (2.17):

$$c_{N,pot} = c_{N,\alpha}^c + c_{N,\alpha}^{nc} + c_{N,q}^{nc}. \quad (2.19)$$

Chordwise Force

The chordwise force c_C for attached flow is simply computed from the normal force:

$$c_{C,pot} = c_{N,pot} \tan(\alpha_e + \alpha_0). \quad (2.20)$$

Pitching Moment

Similarly to the normal force, the pitching moment expression is built from the four different indicial response functions:

$$c_{M,pot} = c_{M_0} + \alpha \cdot \left(\left[\frac{1}{4} - x_{ac} \right] \frac{c_{L,\alpha}}{\beta} \phi_{M,\alpha}^c - \frac{1}{M} \phi_{M,\alpha}^{nc} \right) + q \cdot \left(-\frac{c_{L,\alpha}}{16\beta} \phi_{M,q}^c - \frac{7}{12M} \phi_{M,q}^{nc} \right). \quad (2.21)$$

The first term denotes the zero-lift moment coefficient. In the second term, the (non-dimensionalized) location of the aerodynamic center x_{ac} was used, because in this arrangement the indicial, circulatory response to a change in angle of attack is identical to the corresponding normal force response:

$$\phi_{M,\alpha}^c = \phi_{N,\alpha}^c. \quad (2.22)$$

2.1. THE AERODYN DYNAMIC STALL MODEL

Formally, the three other indicial functions for the pitching moment read:

$$\phi_{M,\alpha}^{nc} = A_3 \cdot \exp\left(-\frac{s}{b_3 T_{M,\alpha}}\right) + A_4 \exp\left(-\frac{s}{b_4 T_{M,\alpha}}\right), \quad (2.23)$$

$$\phi_{M,q}^c = 1 - A_5 \cdot \exp(-b_5 \beta^2 s) \quad (2.24)$$

and

$$\phi_{M,q}^{nc} = \exp\left(-\frac{s}{T_{M,q}}\right). \quad (2.25)$$

In the pitching moment equation, six additional empirical constants (A_3 , A_4 , A_5 , b_3 , b_4 and b_5) as well as two time constants $T_{M,\alpha}$ and $T_{M,q}$ appear:

$$T_{M,\alpha} = \frac{2U}{a} \cdot \frac{1}{(1-M) + \frac{c_{L,\alpha}}{2} M^2 \beta (A_1 b_1 + A_2 b_2)} \quad (2.26)$$

and

$$T_{M,q} = \frac{2U}{a} \cdot \frac{1}{(1-M) + c_{L,\alpha} M^2 \beta (A_1 b_1 + A_2 b_2)}. \quad (2.27)$$

However, in AeroDyn, not all of the terms are computed. This is most likely due to the fact that the pitching moment for attached flow term is actually not required. The three terms that are computed will be used for the pitching moment for separated flow at a later stage, as shown in the subsequent section.

The non-circulatory pitching moment response to a change in angle of attack $c_{M,\alpha}^{nc}$ can be expressed in terms of the non-circulatory normal force response:

$$c_{M,\alpha}^{nc}(s_i) = -\frac{1}{4} c_{N,\alpha}^{nc}(s_i) \quad (2.28)$$

found in Equation (2.15). The non-circulatory pitching moment response due to a change in pitch rate is approximated as follows:

$$c_{M,q}^{nc}(s_i) = -\frac{1}{4} c_{N,q}^{nc} - \frac{c}{3aM} \left(\frac{1}{(1-M) + \pi M^2 \beta (A_1 b_1 + A_2 b_2)} \right)^2 \cdot \left(\frac{q(s_i) - q(s_{i-1})}{\Delta s} - K_q''(s_i) \right). \quad (2.29)$$

It also contains a term involving the corresponding normal force response $c_{N,q}^{nc}$ (from Equation (2.17)) and besides that, a term constituting of a deficiency function

2.1. THE AERODYN DYNAMIC STALL MODEL

K_q'' defined as:

$$K_q''(s_i) = K_q''(s_{i-1}) \cdot \exp\left(-\frac{c}{ak_{M,q}^2} \Delta s\right) + \frac{q(s_i) - q(s_{i-1})}{\Delta s} \cdot \exp\left(-\frac{c}{2ak_{M,q}^2} \Delta s\right) \quad (2.30)$$

where

$$k_{M,q} = \frac{7}{15((1-M) + 1.5c_{L,\alpha}A_5b_5\beta M^2)}. \quad (2.31)$$

Values for the constants are suggested as $A_5 = 1$ and $b_5 = 5$. Finally, the circulatory pitching moment response due to a change in pitch rate q reads:

$$c_{M,q}^c(s_i) = -\frac{c}{16\beta U_{res}} c_{L,\alpha} \left(\frac{q(s_i) - q(s_{i-1})}{\Delta s} - K_q'''(s_i) \right), \quad (2.32)$$

in which K_q''' is another deficiency function expressed as:

$$K_q'''(s_i) = K_q'''(s_{i-1}) \cdot \exp(-b_5\beta^2 \Delta s) + A_5 \frac{q(s_i) - q(s_{i-1})}{\Delta s} \cdot \exp\left(-b_5\beta^2 \frac{\Delta s}{2}\right). \quad (2.33)$$

2.1.2 Unsteady Partially Separated Flow (Submodel)

When it comes to separated flow over thick airfoils, two different types can be distinguished: separation at the leading edge, which is stall in the classical sense and will be treated in Section 2.1.3, or at the trailing edge, which can be considered partial stall, because only a fraction of the airfoil's surface is subjected to separated flow. Leading edge stall will lead to fully separated flow over the upper surface accompanied by vortices shedding from the leading edge and travelling over the upper surface. Besides that, separation emanating from the trailing edge is another possible manifestation of stall, characterized by the fact that, once flow reversal in the boundary layer occurs, it will not automatically spread upstream up to the leading edge. Instead, the flow rather remains attached from the leading edge up to the separation point and the separation will only move upstream with increasing angle of attack, but remains stable otherwise. The degree of separation is characterized by the separation point location along the chord.

2.1. THE AERODYN DYNAMIC STALL MODEL

Trailing edge separated flow is modelled according to Kirchhoff's theory for lift and drag of a flat plate at angle of attack α , from which expressions for normal and chordwise force have been derived:

$$c_N = c_{L,\alpha} \cdot (\alpha - \alpha_0) \cdot \frac{1}{4} \cdot (1 + \sqrt{f})^2. \quad (2.34)$$

and

$$c_c = c_{L,\alpha} \cdot (\alpha - \alpha_0) \cdot \sqrt{f} \cdot \tan(\alpha). \quad (2.35)$$

Even though it is based on two-dimensional potential theory, the extension of an additional free streamline allows a separation point to exist at a particular location along the leeward surface. The location, denominated as f (cf. Fig. 2.2) was expressed in terms of relative chord, such that $f = 0.0$ represents separation occurring at the leading edge (corresponding to fully stalled flow), $f = 0.25$ at the quarter-chord point and $f = 1.0$ at the trailing edge (corresponding to fully attached flow). If the lift curve of static experiments is available, then f can be related to α . Figure 2.3 shows this relationship (f over α in full line) for a particular airfoil (S809, $Re = 1,000,000$, static).

Figure 2.2: Separated flow over flat plate according to Kirchhoff theory

The dynamic stall model in AeroDyn uses such static data to determine the separation point location depending on α . The data points are generated from Equation (2.36), a rearrangement of Equation (2.34):

$$f = \left(2 \sqrt{\frac{c_N}{c_{L,\alpha} \cdot \alpha}} - 1 \right)^2 := f_N. \quad (2.36)$$

By interpolation of data points, a relationship between α and the separation point location can be created.

AeroDyn actually uses two such relationships, one for the normal force (q. (2.36))

2.1. THE AERODYN DYNAMIC STALL MODEL

Figure 2.3: Separation point location f_N and its delayed counterparts f'_N and f''_N

and another derived from the chordwise force expression (Eq. (2.35)). The resulting equation reads:

$$f = \left(\frac{c_c}{c_{L,\alpha} (\alpha - \alpha_0) \tan(\alpha)} \right)^2 := f_c, \quad (2.37)$$

To distinguish the two quantities, they are denoted f_N and f_c .

The original Beddoes-Leishman model incorporated only one f – derived from the lift curve. Besides that, it employed a piecewise approximation of the data points via two (later: three) exponential functions – an approach, which is considerably more inaccurate, but yields smoother results in the stalled regime.

Normal Force

Equation (2.34) has been derived from potential theory, so it is based on steady flow. In order to account for unsteady effects, the separation location undergoes a sequence of modifications. First, the normal force (from the attached flow section)

2.1. THE AERODYN DYNAMIC STALL MODEL

is delayed into a so-called *effective normal force coefficient* c'_N :

$$c'_N = c_{N,pot} - D_p \quad (2.38)$$

via a deficiency function D_p :

$$D_p(s_i) = D_p(s_{i-1}) \cdot \exp\left(-\frac{\Delta s}{T_p}\right) + (c_{N,pot}(s_i) - c_{N,pot}(s_{i-1})) \cdot \exp\left(-\frac{\Delta s}{2T_p}\right). \quad (2.39)$$

This deficiency function is supposed to account for temporal effects on the pressure distribution. It contains an empirical time constant T_p , denoted in the AeroDyn manual as “boundary layer, leading edge pressure time constant” [20]. From the effective normal force c'_N , an *effective angle of attack* is computed according to:

$$\alpha_f = \frac{c'_N}{c_{L,\alpha}} + \alpha_0. \quad (2.40)$$

A new estimate for the separation point location is then made based on this effective angle of attack. In AeroDyn, this is achieved via interpolation of f_N from static data look-up tables with α_f as input as mentioned above, whereas the original Beddoes-Leishman model used piecewise exponential approximating functions. This new estimate f'_N , which can be considered delayed with respect to f_N , is subsequently delayed in another operation into the effective separation point f''_N :

$$f''_N = f'_N - D_{f_N} \quad (2.41)$$

The corresponding deficiency function D_{f_N} is defined as:

$$D_{f_N}(s_i) = D_{f_N}(s_{i-1}) \cdot \exp\left(-\frac{\Delta s}{T_f}\right) + (f'_N(s_i) - f'_N(s_{i-1})) \cdot \exp\left(-\frac{\Delta s}{2T_f}\right). \quad (2.42)$$

The lag associated with D_{f_N} is supposed to model the delayed boundary layer response to a change in angle of attack or pitch rate, quantified by the time constant T_f , which may change during the simulation. Depending on the state of the flow (attached flow, attached with trailing edge separation, fully separated), a modification of T_f is supposed to model these different states as well as certain phenomena

2.1. THE AERODYN DYNAMIC STALL MODEL

such as accelerated separation point movement or slowed reattachment. This is realized in the actual program as follows: upon launching the program, the model is activated with a certain initial value T_{f0} , from which the actual value of T_f will be computed by multipliers that are set according to the situation. The delayed effective separation point location f_N'' is then used in the Kirchhoff expression for the normal force (Eq. (2.34)), it represents the circulatory component of the normal force. The total normal force is then summed up from the two non-circulatory normal force responses (due to change in angle of attack and pitch rate) $c_{N,\alpha}^{nc}$ and $c_{N,q}^{nc}$ from the previous section and the Kirchhoff component:

$$c_N = c_{N,\alpha}^{nc} + c_{N,q}^{nc} + c_{L,\alpha} \cdot \alpha_e \cdot \frac{1}{4} \left(1 + \sqrt{f_N''} \right)^2. \quad (2.43)$$

Comparing this equation to the expression for the normal force under attached flow (Eq. (2.11)), the Kirchhoff component of the normal force can be interpreted as the circulatory component of the normal force modified by the factor $\frac{1}{4} \left(1 + \sqrt{f_N''} \right)^2$ to account for viscous and unsteady effects.

Chordwise Force

The method for computing the chordwise force begins with the effective angle of attack α_f from Equation (2.40), from where it follows the approach for the normal force: α_f is used to look up an effective separation point location f_C' (by plugging α_f into Equation (2.37), which gets delayed into f_C'' according to:

$$f_C'' = f_C' - D_{f_C}. \quad (2.44)$$

The deficiency function D_{f_C} is identical to the deficiency function of the normal force:

$$D_{f_C}(s_i) = D_{f_C}(s_{i-1}) \cdot \exp\left(-\frac{\Delta s}{T_f}\right) + (f_C'(s_i) - f_C'(s_{i-1})) \cdot \exp\left(-\frac{\Delta s}{2T_f}\right). \quad (2.45)$$

The chordwise force c_C can then be computed as

$$c_C = c_{C,pot} \left(\sqrt{f_C''} \right). \quad (2.46)$$

2.1. THE AERODYN DYNAMIC STALL MODEL

Pitching Moment

The pitching moment expression contains the three indicial response terms from the attached flow section (equations (2.28), (2.29) and (2.32)). As the fourth component – corresponding to the circular indicial response to a change in angle of attack – a term is added ($c_M(\alpha'_f)$) that is simply built from looking up a delayed effective angle of attack in static pitching moment data, so that the equation for the total moment reads:

$$c_M = c_M(\alpha'_f) + c_{M,q}^c + c_{M,\alpha}^{nc} + c_{M,q}^{nc} \quad (2.47)$$

This delayed effective angle of attack α'_f is computed from α_f after subtracting a deficiency function D_{α_f}

$$\alpha'_f = \alpha_f - D_{\alpha_f} \quad (2.48)$$

which is equal in structure to D_{f_N} and D_{f_C} , except for the factor 0.1 applied to the time constant T_f :

$$D_{\alpha_f}(s_i) = D_{\alpha_f}(s_{i-1}) \cdot \exp\left(-\frac{\Delta s}{0.1T_f}\right) + (\alpha_f(s_i) - \alpha_f(s_{i-1})) \cdot \exp\left(-\frac{\Delta s}{0.2T_f}\right). \quad (2.49)$$

This will make the equation for the pitching moment read:

$$\begin{aligned} c_M(s_i) = & c_M(\alpha'_f, s_i) - \frac{c}{16\beta U_{res}} c_{L,\alpha} \left(\frac{q(s_i) - q(s_{i-1})}{\Delta s} - K_q'''(s_i) \right) \\ & - \frac{T_{N,\alpha}}{M} \left(\frac{\alpha(s_i) - \alpha(s_{i-1})}{\Delta s} - K'_\alpha(s_i) \right) + \frac{T_{N,q}}{4M} \left(\frac{q(s_i) - q(s_{i-1})}{\Delta s} - K'_q(s_i) \right) \\ & - \frac{c}{3aM} \left(\frac{1}{(1-M) + \pi M^2 \beta (A_1 b_1 + A_2 b_2)} \right)^2 \\ & \cdot \left(\frac{q(s_i) - q(s_{i-1})}{\Delta s} - K_q''(s_i) \right). \end{aligned} \quad (2.50)$$

2.1.3 Unsteady Fully Separated Flow (Submodel)

Full separation occurs, when the airfoil's suction is not sufficient enough for the fluid to sustain the flow around the curvature of the leading edge. In the case of a static airfoil, a *critical angle of attack* is associated to this event. The “critical”

2.1. THE AERODYN DYNAMIC STALL MODEL

normal force or lift coefficient persisting at this particular angle serves as an indicator for stall to occur.

Evans and Mort [21] developed a simple, yet feasible criterion to predict leading edge separation under dynamic conditions, according to which they identified the leading edge pressure and pressure gradient to trigger the onset of flow separation. Later, Beddoes [22, 23] found that the leading edge pressure as a sole criterion was sufficient. Since the leading edge pressure is correlated to the normal force, c_N can then also be taken as criterion for leading edge separation in the dynamic case, so it can be said that there exists a critical normal force under dynamic conditions as well. Another feature of this relation is the fact that the leading edge pressure is delayed with respect to the normal force, which can comfortably be modelled by a deficiency function. In fact, the deficiency function of Equation (2.38) is used for this. By comparing this delayed normal force, the effective normal force c'_N , to a critical normal force, a criterion for stall to occur has been established, stating that stall occurs whenever:

$$c'_N > c_{N1}. \quad (2.51)$$

Here, c_{N1} is the critical normal force at which static stall occurs. In order to allow for stall to also be detected at negative angles of attack, the normal force is checked in a second condition whether it falls below a critical value c_{N2} :

$$c'_N < c_{N2}. \quad (2.52)$$

Cambered airfoils in general have a non-symmetric lift curve, so the threshold for negative angle of attack c_{N2} should not simply have the negative value of c_{N1} . This is the reason for a second, distinct condition on c'_N .

Once dynamic stall has been triggered by Equations (2.51) or (2.52), a variable τ_V to track the vortex travelling over the airfoil surface is created: it counts the time for a vortex to pass from leading to trailing edge. Another variable, or rather: a time constant, T_{VL} represents the time span it takes for a vortex to move over the airfoil from leading to trailing edge. With these two variables, τ_V and T_{VL} , the vortex shedding process triggered by leading edge separation will be described in AeroDyn. It continues as long as:

$$\tau_V(s_i) < T_{VL}. \quad (2.53)$$

2.1. THE AERODYN DYNAMIC STALL MODEL

Normal Force

The onset of dynamic stall will cause the leading edge vortex to be generated from the rolled up shear layer. An additional normal force term c_N^v to account for the presence of the leading edge vortex is realized via the following equation:

$$c_N^v(s_i) = c_N^v(s_{i-1}) \cdot \exp\left(-\frac{\Delta s}{T_V}\right) + (c_V(s_i) - c_V(s_{i-1})) \cdot \exp\left(-\frac{\Delta s}{2T_V}\right), \quad (2.54)$$

which is simply added on top of the normal force. The time constant T_V appearing in this equation is another empirical value. Similar to T_f , its value is computed from an initial value T_{V0} and may be adjusted during simulation. It may be increased in value to account for accelerated vortex delay, for example. The vortex-induced component of the normal force term c_N^v has a structure that is often used in Beddoes-Leishman-type models, incorporating the decaying character via an exponential term (the first term) complemented by a producing term (the second term) due to the accumulated leading edge vorticity c_V . So, as long as the second term – caused by a change in c_V – keeps generating “vorticity”, c_N^v will continue to contribute to the normal force. In case this should recede (e.g. due to the kinematics of the blade), the contribution of c_N^v will naturally decay to zero. Moreover, c_N^v will also decay to zero once the time counter τ_V exceeds T_{VL} , i.e. once the vortex has passed the trailing edge of the blade, because then, the second term will be set to zero by the AeroDyn algorithm. c_V is computed as the difference between the circulatory normal force components of attached flow ($c_{N,\alpha q}^c = c_{L,\alpha} \cdot \alpha_e$, Eq. (2.11)) and separated flow ($c_{L,\alpha} \cdot \alpha_e \cdot \frac{1}{4} (1 + \sqrt{f_N''})^2$, Eq. (2.43)):

$$c_V = c_{L,\alpha} \cdot \alpha_e \cdot \left[1 - \frac{1}{4} \left(1 + \sqrt{f_N''} \right)^2 \right]. \quad (2.55)$$

Chordwise Force

The tangential force is increased by a vortex-induced component c_C^v that reads:

$$c_C^v = c_N^v \cdot \tan(\alpha_e) \cdot \left(1 - \frac{\tau_V}{T_{VL}} \right). \quad (2.56)$$

Its effect is of limited time, represented by the term $1 - \frac{\tau_V}{T_{VL}}$, a linear decrease in τ_V . T_{VL} denotes the time it takes for the leading edge vortex to travel over the airfoil.

2.2. EXPERIMENTAL DATA AND TEST CASES

Pitching Moment

The pitching moment is affected by the leading edge vortex travelling over the airfoil, the induced normal force increment translates into a additional pitching moment component c_M^v , which is constructed as the product of c_N^v , the normal force increment, $x_{cp} - 0.25$, the (non-dimensionalized) distance between center of pressure and quarter-chord, and a term describing the movement of the center of pressure due to the influence of the moving vortex:

$$c_M^v = -(x_{cp} - 0.25) \cdot \left(1 - \cos\left(\frac{\pi \tau_V}{T_{VL}}\right) \right) \cdot c_N^v. \quad (2.57)$$

2.2 Experimental Data And Test Cases

The necessary data for providing look-up tables (static lift and pitching moment) as well as error computing was taken from the wind tunnel investigations at Ohio State University during the 1990s, since the corresponding reports are publicly available. This project – performed at Ohio State University on behalf of the National Renewable Energy Laboratory – involved the measurement of more than 10 wind turbine-related airfoils under steady and unsteady conditions over a range of Reynolds numbers from 750,000 to 1,500,000. Among those 10 airfoils are the S809 [24], the S825 [25] and the LS(1)-0417MOD [26, 27] that appear in this work. In the dynamic cases, the particular airfoil performs sinusoidal pitching motions, such that the pitch angle follows the relationship:

$$\alpha(t) = \alpha_{mean} + \alpha_{amp} \cdot \sin(2\pi\kappa t). \quad (2.58)$$

The mean angle α_{mean} was chosen to be 8.0° , 14.0° and 20.0° . Oscillating amplitudes were either $\alpha_{amp} = 5.5^\circ$ or 10.0° . The oscillating frequencies had the values $\omega = 0.6Hz$, $1.2Hz$ and $1.8Hz$, which – depending on the Reynolds number – translates into different (non-dimensional) frequencies κ ($\kappa = \frac{c}{2U_{res}}\omega$). The possible combination of parameters span a set of 18 cases, out of which 8 were used as test cases in this work. These chosen cases represent the extreme ends (highest and lowest) of the spectrum of kinematic conditions investigated in the Ohio State University campaign in terms of oscillating frequencies, mean pitch angle and pitch amplitudes. Table 2.1 gives an overview of these cases.

Even though numerical models rely on real world data for validation and/or calibration, results from experimental campaigns should not be blindly trusted or

2.2. EXPERIMENTAL DATA AND TEST CASES

amplitude angle α_{amp}	mean angle α_{mean}	
	8.0°	20.0°
5.5°	TC101 $\kappa = 0.026$	TC103 $\kappa = 0.026$
	TC105 $\kappa = 0.077$	TC107 $\kappa = 0.077$
10.0°	TC106 $\kappa = 0.026$	TC108 $\kappa = 0.026$
	TC102 $\kappa = 0.078$	TC104 $\kappa = 0.078$

Table 2.1: Overview of kinematics of dynamic test cases

Figure 2.4: Kinematics of test cases

considered as “fact”. No matter how much effort put into establishing controlled conditions, chances for unwanted (or even unknown) effects to affect the outcome of experiments always exist. In the case of one of the airfoils investigated in this work, the S809, there are – besides the Ohio State University campaign [24] – other sources of experimental data: the S809 was also tested at Glasgow University by Sheng et al. [28], at Colorado State University by Butterfield et al. [29] and at TU Delft (by Somers [30]). Comparing results from these experimental campaigns, several differences can be observed: lift and drag curves – even in the attached flow regime at low angles of attack – show numerous deviations. The Glasgow

2.2. EXPERIMENTAL DATA AND TEST CASES

University lift curve e.g. leaves the linear regime a few degrees earlier than the two others. While Butterfield et al. [29] mention the static stall angle – albeit at a lower Reynolds number – to be at 18° , Sheng et al. [28] recognized it at 19.16° . Zero lift has been recorded in a rang between -0.64° and -1.18° . From these discrepancies in the recorded observations it can be concluded that experimental data should be taken “with a pinch of salt”.

2.2.1 Transformation Of Loads Into From α Into Time Space

The experimental data taken from the Ohio State University campaign [24] is extracted from plots in the α -load (α - c_L , α - c_D and α - c_M) space, i.e. from graphs plotting the load over the angle of attack. Data produced by the dynamic stall model can be output in this space, too. However, it is not suitable for comparison, because in the α -load space, there is no unique mapping from α to the load, which can easily be seen from the graphs in which the results are usually presented (cf. Fig. 2.5): due to the periodic motion of the airfoil, there are two data points for each angle of attack (except upper and lower dead center, of course). In mathemat-

Figure 2.5: Lift force over angle of attack α

ical terms, the trigonometric function describing the airfoil’s pitching motion (sine or cosine function) is *not injective*. That is why the loads extracted from graphs of experimental data need to be transformed into time-load space.

Recalling the original equation describing the pitching motion of the airfoil (2.58),

2.2. EXPERIMENTAL DATA AND TEST CASES

Figure 2.6: Lift force over time t

the mathematical relationship between time t and angle of attack α was defined as:

$$\alpha(t) = \alpha_{mean} + \alpha_{amp} \cdot \sin(2\pi\kappa t).$$

Inverting this equation, an expression for the time can be found:

$$t(\alpha) = \frac{1}{2\pi\kappa} \cdot \arcsin\left(\frac{\alpha - \alpha_{mean}}{\alpha_{amp}}\right). \quad (2.59)$$

Since the inverse trigonometric function in Equation (2.59) is limited in its range and only partially covers on full revolution, it needs to be defined piecewise in at least two intervals: upstroke and downstroke. Mathematically, these two cases can be distinguished by the rate of change of α , which is either positive or negative ($\dot{\alpha} \gtrless 0$). The intervals are shifted by a half-period $\frac{T}{2}$ with respect to each other and the sign of the arcsin function will switch from one interval to the next. In order to achieve a time $t = 0s$ at $\alpha = \alpha_{mean}$, the first interval is chosen to begin at α_{mean} . This will make the expression for the time to be computed from the angle of attack

2.3. PERFORMANCE OF THE AERODYN DYNAMIC STALL MODEL

read as follows:

$$t(\alpha) = \begin{cases} \frac{1}{2\pi\kappa} \cdot \arcsin\left(\frac{\alpha - \alpha_{mean}}{\alpha_{amp}}\right) & \forall \quad \dot{\alpha} > 0 \quad \wedge \\ & \alpha_{mean} < \alpha < (\alpha_{mean} + \alpha_{amp}) \\ \frac{-1}{2\pi\kappa} \cdot \arcsin\left(\frac{\alpha - \alpha_{mean}}{\alpha_{amp}}\right) + \frac{T}{2} & \forall \quad \dot{\alpha} \leq 0 \quad \wedge \\ & (\alpha_{mean} - \alpha_{amp}) \leq \alpha \leq (\alpha_{mean} + \alpha_{amp}) \\ \frac{1}{2\pi\kappa} \cdot \arcsin\left(\frac{\alpha - \alpha_{mean}}{\alpha_{amp}}\right) + T & \forall \quad \dot{\alpha} > 0 \quad \wedge \\ & (\alpha_{mean} - \alpha_{amp}) < \alpha < \alpha_{mean} \end{cases} \quad (2.60)$$

With this, it is possible to convert the experimental data given in $c_L(\alpha)$ mapping into $c_L(t)$ (and accordingly for c_D and c_M). Figure 2.6 shows the data from Figure 2.5 converted into t - c_L space. Of course, this step is omitted if experimental data is available in the proper format in the time-load space.

2.3 Performance Of The AeroDyn Dynamic Stall Model

The following sections discuss results of the NREL AeroDyn dynamic stall model when applied to the S809 airfoil according to the test cases of Section 2.2 and with default values (values suggested by the S809 data sheet provided by AeroDyn). Table 2.2 shows these values. The corresponding figures with lift, drag and moment curves can be found in the appendix, i.e. Sections A.1.1 to A.1.8 beginning on page 88.

2.3.1 Test Case 101

The lift results cover the area spanned by the experimental data and match the shape of the hysteresis loop well. The shape of the drag curve is captured qualitatively well, too. The model underpredicts the drag at low angles of attack, while it is the opposite at larger angles. The pitching moment results are difficult to compare, since the experimental data shows a great deal of scatter and no clear shape can be recognized suggesting high sensitivity of c_M . The dynamic stall model results lie within the range of c_M values and it does not suggest a crossing of the hysteresis loop. This is mostly due to the narrow range of c_M values (note the y-axis range

2.3. PERFORMANCE OF THE AERODYN DYNAMIC STALL MODEL

name	default value	used in equation
A_1	0.3	(2.13)
A_2	0.7	(2.13)
b_1	0.14	(2.14)
b_2	0.53	(2.14)
$T_{f0} (T_f)$	3.0	(2.42), (2.45), (2.49)
$T_{V0} (T_V)$	5.0	(2.54)
T_p	1.7	(2.39)
T_{VL}	35.0	(2.53), (2.56), (2.57)
α_0	-0.6391	(2.12), (2.20), (2.34), (2.37), (2.40) (2.35),
η_e	1.0	(2.35)
$c_{L,\alpha}$	6.438	(2.9), (2.10), (2.11), (2.21), (2.26), (2.27), (2.31), (2.32), (2.34), (2.35), (2.36), (2.37), (2.40), (2.43), (2.50)
c_{N1}	0.9292	(2.51)
c_{N2}	-0.6082	(2.52)
c_{D0}	0.0018	(2.2)
$\hat{x}_{cp}$	0.3	(2.57)

Table 2.2: Default values of parameters and time constants for S809 airfoil

from -0.1 to 0.0), whereas the pitching moment in other cases varies over a wider range.

It is striking that the experimental data do not seem to cover the full range of angle of attack. According to the specifications, α should be in the range from 2.5° to 13.5° . However, no information about possible reasons for this was given in the report related to the experimental campaign [24]. This observation can be made throughout other cases as well, e.g. 103, 105 and 107.

2.3.2 Test Case 102

The lift curve of test case 102 is captured satisfactorily, in particular, the pitch up phase until dynamic stall occurs. The delayed stall angle is also hit very well. However, the stalled phase is not well-pronounced, and thus, the returning phase of the pitching, which suggests that the vortex lift might be predicted too small. The drag coefficient is predicted very well except for the sudden shoot-up around 18° , which is not present at all. As for the pitching moment, the results of the model lie within the spanned area of the experimental results.

2.3. PERFORMANCE OF THE AERODYN DYNAMIC STALL MODEL

2.3.3 Test Case 103

The model captures the lift distribution well in the attached flow regime, but during upstroke towards high angles of attack, lift is too low and also almost no hysteresis exists. Also, scatter in the experimental lift values of test case 103 suggests either vibration in the wind tunnel arrangement (for which it is probably too low in frequency) or rather the traversal of the shedded leading edge vortex. This feature is not present in the NREL dynamic stall model. Moreover, the pitch angle range seems misrepresented in the experimental results (as mentioned for test case 101). The model's drag curve gives good results during upstroke, but can not capture the sudden peak (hinting the leading edge vortex) around 23° . The spread between pitch up and pitch down motion is almost non-existent for higher angles. This is also the case for the pitching moment, where negative moment peak from leading edge vortex is not captured.

2.3.4 Test Case 104

The dynamic stall model underpredicts the lift, i.e. it is significantly too low during upstroke and too high during returning motion. The vortex lift peak is not modelled at all. Also, it is unclear whether the crossing of the curve appears in the experimental data or not. Overall, the drag is predicted very well – except for the final stage of the upstroke motion, where height and angle of the peak are missed. The pitching moment is reasonably well modelled. The missing vortex lift peaks from the lift results do not appear here, either. characteristic for peak stall better.

2.3.5 Test Case 105

Overall, the results of test case 105 are modelled very well. This holds for the lift, where only during downstroke, the model slightly overpredicts the wind tunnel data. Modelled drag results give an excellent match, which can also be said to a somewhat lesser extent for the pitching moment.

2.3.6 Test Case 106

The dynamic stall model to capture the lift very well during the entire cycle, which is most likely due to the fact that the motion of test case 106 – like case 105 – occurs mainly within the attached flow regime. Also, the drag is matched very closely, except for an underprediction of the drag peak at lift stall. The pitching moment

2.3. PERFORMANCE OF THE AERODYN DYNAMIC STALL MODEL

curve resembles the experimental data well in terms of where the hysteresis loop is rather thin or where it is further extended.

2.3.7 Test Case 107

The model fails to capture increasing lift during upstroke of test case 107, but rather suggests declining c_L , and the leading edge vortex-generated peak towards the end of upstroke. The returning motion, however, is predicted well. When it comes to drag, a slight overshoot at low angles can be observed, but no drag overshoot during leading edge vortex at all. This is also the case for the pitching moment, where the large drop in c_M fails to be modelled. Instead, two other, seemingly unphysical bumps are present in the moment curve.

2.3.8 Test Case 108

The last test case again shows the lack of lift overshoot from leading edge vortex during upstroke, the downstroke matches the experimental data reasonably. However, it is clear whether the experimental data also feature a crossing of the lift curve. Overall, the qualitative behaviour of lift was captured, though, which also holds true for the drag and pitching moment results: besides the lack of peak loads during stall, the data provided by the model matches the experiments very well.

2.3.9 Discussion

Based on the results of the dynamic test cases, the following statements can be made:

- In some of the cases, the experimental data does not cover the actual range of angle of attack (101, 103, 105, 107). Neither in the reports by Ohio State University [24] nor in literature of other dynamic stall models involving these test cases this is mentioned. The question arises whether the experimental data should be scaled along the horizontal axis to obtain the appropriate α range.
- Lift is overall well-captured, in particular in the attached regime. A common pattern seems to be a slight overshoot during the downstroke. Aside from that, the qualitative features are predicted by the model, except lift peaks caused by the leading edge vortex detachment. The AeroDyn dynamic stall model lacks any feature that could reproduce fluctuating loads

2.4. SENSITIVITY OF PARAMETERS AND TIME CONSTANTS IN THE NREL DYNAMIC STALL MODEL

observed when the leading edge vortex travels over the airfoil. The original Beddoes-Leishman model had such a feature, however.

- The drag is generally well-predicted in both attached and separated flow regime. Apart from some minor flaws, the only issue here also seems to be the lack of drag overshoot at the point of stall as well the subsequent increased drag generated by the detached leading edge vortex.
- Of all the three load types, the moment seems the most difficult to predict properly – this is something that was already mentioned by Leishman ([31], p. 264) – but considering the small range in which the pitching moment is distributed, this impression might be deceptive. Similar to lift and drag, the failure to model the leading edge vortex-generated pitching moment fluctuations is the most significant source of error.

2.4 Sensitivity Of Parameters And Time Constants In The NREL Dynamic Stall Model

In an attempt to qualitatively assess the sensitivity of the parameters and time constants of the AeroDyn dynamic stall model, simulations of all eight test cases with the S809 airfoil have been performed, in which each parameter was individually modified from its default value: one simulation with double and another simulation with half the value. All figures are listed in the subsections of Appendix A.2 rather than here because of the sheer number.

2.4.1 Parameters A_1, A_2, b_1, b_2

Figures for parameters A_1, A_2, b_1 and b_2 can be found in Sections A.2.1, A.2.2, A.2.3 and A.2.4. A_1 and A_2 each generate more hysteresis in lift, drag and moment if its value is increased, with the exception of the lift in cases *TC104* and *TC107*. On the other hand, b_1 and b_2 produce more hysteresis if its value is reduced. The four Parameters are all used to compute deficiency functions X_1 and X_2 (equations (2.13) and (2.14)) necessary for determining the effective angle of incidence α_e . These deficiency functions are supposed to reduce the angle of attack to replicate the effect of delayed build-up of circulation around the airfoil. Lift curves of test cases *TC104* and *TC107* seem to feature somewhat less hysteresis opposing the observed trend, which is most likely because the characteristics of the hysteresis

2.4. SENSITIVITY OF PARAMETERS AND TIME CONSTANTS IN THE NREL DYNAMIC STALL MODEL

curve is different: the lower loop has inverted rotational sense (in the case of lift: clockwise) compared to the other cases (counter-clockwise). Another manifestation of the delaying effect observed is that a modification of b_1 or b_2 can shift the extreme ends of the load curve. This is more visible for b_1 than for b_2 (probably due to the different weighting) and more so in cases with higher frequency, which could hint that this phenomenon is related to inertia effects as well.

2.4.2 Time Constant T_f

Increasing the value of T_f (by choosing a larger initial value T_{f0}) will extend the delay of f'' with respect to f' and f . This will also lead to further delayed response of the loads, which is most clearly visible in the lift results, where the overshoots are strongest. A smaller value will have the opposite effect. All relevant plots can be found in Section A.2.5. Besides that, when increasing T_f , lift tends to go down at low angles and to go up for larger angles. This can be observed in opposite direction, albeit to a much smaller degree, for the drag.

2.4.3 Time Constant T_p

As can be seen from the plots in Section A.2.6, the time constant T_p behaves similar to T_f in the sense that increasing its value will lead to a more lagged response creating more hysteresis. In fact, the graphs of the T_p sensitivity analysis resemble the ones of the T_f analysis, but with less increment. This makes sense as both time constants are used in a delay term of equivalent structure applied to the normal force.

2.4.4 Time Constant T_V

The plots for the time constant T_V can be found in Section A.2.7. T_V affects the result not during the entire simulation, but during the flow regime of vortex lift, i.e. where (build-up of) the leading edge vortex creates excessive lift and drag, manifesting itself in all three loads. A larger T_V value generates more of that additional lift, moment and drag and thus, the overshoots in lift, moment and drag become larger. However, drag and pitching moment are significantly less affected than the lift.

2.4. SENSITIVITY OF PARAMETERS AND TIME CONSTANTS IN THE NREL DYNAMIC STALL MODEL

2.4.5 Time Constant T_{VL}

T_{VL} has no impact on the normal force and hence, only little influence on the lift. Whatever deviation observed in c_L stems from the chordwise force contribution (from Equation (2.56)). It only affects the vortex-induced loads that may occur after leading edge separation. From the corresponding figures (in Appendix A.2.8), it can be deduced that a reduction of T_{VL} will magnify the pitching moment peak. The influence on the chordwise force c_C is not so much manifested in terms of an increase or decrease of the response, but rather for how much time the vortex-induced load component acts. This is due to the fact that the time constant T_{VL} controls the length of the interval of vortex shedding (i.e. the time a shed vortex travels over the airfoil), as mentioned in Section 2.1.3. So, by increasing this time T_{VL} , the additional lift and drag generation can be extended over a longer period.

Chapter 3

Optimization Methods For Finding Optimal Time Constants And Parameters In Beddoes-Leishman Dynamic Stall Models

3.1 Optimization – General Remarks

An optimization problem, stated as: “given a range of certain variables x , find those particular values x that maximize (or minimize) the value of an objective that depends on x ”, can mathematically be formulated in the following way:

$$\begin{aligned} & \underset{\underline{x}}{\text{minimize}} && g(\underline{x}) \\ & \text{subject to} && \underline{h}(\underline{x}) \leq \underline{0} \\ & && \underline{i}(\underline{x}) = \underline{0} \end{aligned}$$

In this expression, also called *standard form*, the vector $\underline{x}$ ($\underline{x} \in \mathbb{R}^n$) denotes the *design variables* of the problem, for which a particular set of values is sought that will minimize the *objective function* g ($g : \mathbb{R}^n \rightarrow \mathbb{R}$). Since minimizing $g(\underline{x})$ is equivalent to maximizing $-g(\underline{x})$, it is not relevant which kind of optimization problem we are dealing with: a minimization problem can simply be converted into a maximization problem by applying a factor of (-1) . The design variables may

3.2. OPTIMIZATION OF PARAMETERS – OCCURRENCES IN LITERATURE

be subject to optional *constraints* $\underline{h}$ and/or $\underline{i}$ ($\underline{h}, \underline{i} \in \mathbb{R}^m$): either in inequality (i.e. $h_i(\underline{x}) \leq 0$) or equality form (i.e. $h_i(\underline{x}) = 0$) and either in implicit (i.e. $h_i(\underline{x}) \leq 0$) or explicit form (e.g. $x_j \leq x_{max}$).

3.2 Optimization Of Parameters – Occurrences In Literature

As of today, there are only two documented instances where proper optimization methods instead of 'trial & error' have been applied to find suitable values for time constants or other empirical parameters in dynamic stall models of Beddoes-Leishman type.

Gonzalez and Munduate of the Spanish National Renewable Energy Centre (CENER) have developed a Beddoes-Leishman type dynamic stall model named DYSTOOL [13], which they used to investigate the NREL UAE phase VI turbine in parked state [14]. In these works, they claim to have employed a genetic algorithm to set time constant T_f and the effective normal force coefficient c'_N . In particular, they state: *"The values were fixed using an optimization based on genetic algorithms, considering the physical sense of the results, and taking into account the values found in the articles of J.G. Leishman and T.S. Beddoes [...]."* [14]. However, no further detail about the method was mentioned, so it is unclear which specific genetic algorithm was employed, what parameters were used, etc. The fact that this was neither mentioned nor it was not published anywhere else may lead to the conclusion that the optimization aspect was not the focus of their work and that consecutively not much effort went into it.

Mert [10] used the Method of Moving Averages (MMA) to find suitable values for time constants in DYNSTALL, the dynamic stall model developed at FFA [12]. This method will be described in more detail in the following Section 3.5.

3.3 Modelling 'Finding Optimal Parameters Of A Dynamic Stall Method' As An Optimization Problem

Setting up an optimization problem to find optimal values for parameters of a dynamic stall model, the design variables obviously need to be these particular parameters. They may or may not be restricted by constraints. An example would be non-negativity of the time constants, which would make sense mathematically

3.4. ERROR COMPUTATION AND SCALARIZING

– they appear in the exponent of an e function and changing the sign would fundamentally change a decaying into a growing character – as well as physically, since negative time periods are impossible. But in general, none of the parameters are limited in any way. The crucial part is the objective function and there are several options on how to set it up. Mert [10] suggested to minimize the error in maximum lift, the error in mean lift curve slope or the error in lift, but then settled for combined error of lift and drag. The error is supposed to be the difference between modelled and experimentally observed quantity. The most holistic choice rather seems to be the difference between all of the modelled and experimentally observed loads, i.e. the error of the three modelled loads, that is supposed to be minimized, simply because we are interested in best possible estimates for all three loads. The dynamic stall model will provide lift, drag and pitching moment as output and, after having computed three errors out of that, there will still be three values instead of one, which can not be used as objective value in an optimization algorithm, because the objective value needs to be scalar. This issue will be addressed in Section 3.4.

Whenever an evaluation of the objective function $g(\underline{x})$ is required, the dynamic stall model needs to be executed with the appropriate set of parameters $\underline{x}$. It will return an error value (by subtracting computed loads from stored experimental data) serving as g . Figure 3.1 visualizes this coupling.

Figure 3.1: Coupling of optimization algorithm and dynamic stall model

Before introducing suitable optimization algorithms, Svandberg's Method of Moving Averages previously mentioned will be described and it will be shown why it is not a suitable method for finding optimal parameters for a dynamic stall model.

3.4 Error Computation And Scalarizing

A dynamic stall model provides estimates of lift, drag and pitching moment at every time step, so the output consists of three vectors. Comparing to experimental data yields errors in these three quantities, each of them also vectors. Now, for an optimization algorithm to be applied to a dynamic stall model, the errors need

3.4. ERROR COMPUTATION AND SCALARIZING

to be mapped into one scalar value, because the objective value of an optimization problem necessarily is a scalar function. Computing the norm of each vector and thereby mapping the three error vectors to three scalars is an appropriate and straightforward solution for one of two issues. To rephrase this in a mathematical manner, let the errors in lift, drag and moment produced by the dynamic stall routine at time step t_i be denoted by $e_L(t_i)$, $e_D(t_i)$ and $e_M(t_i)$ according to:

$$e_L(t_i) = |c_{L, sim}(t_i) - c_{L, exp}(t_i)|, \quad (3.1)$$

$$e_D(t_i) = |c_{D, sim}(t_i) - c_{D, exp}(t_i)|, \quad (3.2)$$

and

$$e_M(t_i) = |c_{M, sim}(t_i) - c_{M, exp}(t_i)|. \quad (3.3)$$

By computing the norm, the errors in vector form become scalars:

$$\epsilon_L = \sqrt{\sum_i [e_L(t_i)]^2}, \quad (3.4)$$

$$\epsilon_D = \sqrt{\sum_i [e_D(t_i)]^2} \quad (3.5)$$

and

$$\epsilon_M = \sqrt{\sum_i [e_M(t_i)]^2}. \quad (3.6)$$

For mapping the three individual errors into one overall or total error, several ways exist. Such a total error ϵ_{total} could, for example, be computed by summing up absolute values or norms of the particular errors:

$$\epsilon_{total} = |\epsilon_L| + |\epsilon_D| + |\epsilon_M| \quad (3.7)$$

to arrive at a scalar value representing an overall error that can serve as objective value to be minimized. However, since the values of lift are usually an order larger than the drag or moment values, which also holds true for the errors, this total error would be dominated by the error in lift. That, in turn, would mean an optimization algorithm would prefer to reduce the error in lift over the other two, because a larger reduction of the objective value can be gained. It would be biased towards

3.5. SVANDBERG'S METHOD OF MOVING ASYMPTOTES

the lift. This issue reflects the fact that the problem is actually a *multi-objective* optimization – i.e. reducing the error in lift, in drag and in moment at the same time. All *Pareto-optimal* solutions (solutions that can not be improved in one parameter without deteriorating another) would be equally good and hence, a subjective preference would be required anyway.

An improvement over simply adding norms is to include weighting factors to scale the errors before adding them up, for which the equation will look like this:

$$\epsilon_{total} = w_L \cdot |\epsilon_L| + w_D \cdot |\epsilon_D| + w_M \cdot |\epsilon_M|. \quad (3.8)$$

But then the question arises how the user is supposed to choose appropriate values for the weights. The errors are not known beforehand, so scaling with weights involving $\epsilon_{L/D/M}$ a priori will not help. Instead, an a posteriori scaling is advisable. One such scaling is by the complementary terms, which means the lift error ϵ_L is scaled by the sum of drag and moment error $\epsilon_D + \epsilon_M$, the error in drag ϵ_D by the sum of lift and moment error $\epsilon_L + \epsilon_M$ and the pitching moment error ϵ_M is scaled by the sum of the errors in lift and drag $\epsilon_L + \epsilon_D$:

$$\epsilon_{total} = \frac{|\epsilon_D| + |\epsilon_M|}{|\epsilon_L| + |\epsilon_D| + |\epsilon_M|} \cdot |\epsilon_L| + \frac{|\epsilon_L| + |\epsilon_M|}{|\epsilon_L| + |\epsilon_D| + |\epsilon_M|} \cdot |\epsilon_D| + \frac{|\epsilon_L| + |\epsilon_D|}{|\epsilon_L| + |\epsilon_D| + |\epsilon_M|} \cdot |\epsilon_M|. \quad (3.9)$$

The term $|\epsilon_L| + |\epsilon_D| + |\epsilon_M|$ in the denominator has been added as a uniform down-scaling, a normalization.

3.5 Svandberg's Method Of Moving Asymptotes

Svandberg has devised the so-called 'Method Of Moving Asymptotes' [11], a convex approximation optimization method for non-linear programming problems. He writes such kind of optimization problem in the following form:

$$\begin{aligned} P: \quad & \text{minimize} \quad g_0(\underline{x}) \\ & \text{subject to} \quad g_i(\underline{x}) \leq \hat{g}_i \\ & \quad \quad \quad x_{j,min} \leq x_j \leq x_{j,max} \quad , \end{aligned} \quad (3.10)$$

3.5. SVANDBERG'S METHOD OF MOVING ASYMPTOTES

in which the vector of design variables is given as $\underline{x} \in \mathbb{R}^n$ (i.e. $\underline{x} = [x_1, x_2, \dots, x_n]^T$). The objective function g_0 and constraints g_i ($i = 1, 2, \dots, m$) are both denoted by the letter g . In contrast to the general notation of Section 3.1, the constraint inequalities here have a right-hand side term $\hat{g}_i$ ($i = 1, 2, \dots, m$), i.e. the absolute term is not included on the left-hand side.

The solution to such a problem can be found via an iterative algorithm:

step 0 choose initial guesses for the design variables $\underline{x}^{(k)}$ and set the iteration counter $k = 0$

step 1 given design variables $\underline{x}^{(k)}$, compute $g_i(\underline{x}^{(k)})$ and $\nabla g_i(\underline{x}^{(k)})$ for $i = 1 \dots m$

step 2 generate the subproblem $P^{(k)}$ from P by approximating the implicit functions g_i via explicit functions $g_i(\underline{x}^{(k)})$

step 3 solve $P^{(k)}$ via the dual problem $D^{(k)}$ and let the optimal solution be the next iteration's design variables $\underline{x}^{(k+1)}$; let $k = k + 1$; go to step 1

The iteration is stopped once a convergence criteria (set up by the user) is satisfied. According to step 2, the objective and (usually implicit) constraint functions are approximated by explicit functions:

$$g_i(x_j) \approx g_i^{(k)}(x_j) = r_i^{(k)} + \sum_{j=1}^n \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right), \quad (3.11)$$

in which

$$p_{ij}^{(k)} = \begin{cases} \frac{\partial g_i}{\partial x_j} (U_j^{(k)} - x_j^{(k)})^2 & \text{if } \frac{\partial g_i}{\partial x_j} > 0 \\ 0 & \text{if } \frac{\partial g_i}{\partial x_j} \leq 0 \end{cases}, \quad (3.12)$$

$$q_{ij}^{(k)} = \begin{cases} 0 & \text{if } \frac{\partial g_i}{\partial x_j} \geq 0 \\ -\frac{\partial g_i}{\partial x_j} (x_j^{(k)} - L_j^{(k)})^2 & \text{if } \frac{\partial g_i}{\partial x_j} < 0 \end{cases} \quad (3.13)$$

and

$$r_i^{(k)} = g_i(x_j^{(k)}) - \sum_{j=1}^n \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - L_j^{(k)}} \right). \quad (3.14)$$

3.5. SVANDBERG'S METHOD OF MOVING ASYMPTOTES

The coefficients p_{ij} and q_{ij} are determined based on the sign of the objective function's and constraints' gradients $\frac{\partial g_i}{\partial x_j}$. Together with the parameters L and U (satisfying $L_j^{(k)} < x_j^{(k)} < U_j^{(k)}$), this makes the approximation $g_i^{(k)}(x_j)$ essentially a linearization of g_i ($i = 0, 1, 2, \dots, m$) in terms of $\frac{1}{x_j}$. Hence, the L and U are called 'moving asymptotes', they are to be set by the user and readjusted every iteration. Depending on the user's choice – closer to the iteration point $\underline{x}^{(k)}$ or further away – the approximations of g become more or less conservative affecting the convergence rate of the optimization algorithm. The purpose of the moving asymptotes is to make the subproblem $P^{(k)}$ convex and thereby to guarantee a solution according to the suggested method, which involves solving the so-called dual problem.

Thus, the subproblem $P^{(k)}$ to be solved every iteration k reads:

$$\begin{aligned} P^{(k)}: \quad & \text{minimize} \quad \sum_{j=1}^n \left(\frac{p_{0j}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{0j}^{(k)}}{x_j - L_j^{(k)}} \right) + r_0^{(k)} \\ & \text{subject to} \quad \sum_{j=1}^n \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right) + r_i^{(k)} \leq \hat{g}_i \\ & \quad \quad \quad x_{j,min} \leq x_j \leq x_{j,max}. \end{aligned} \quad (3.15)$$

In his article, Svanberg [11] describes how to use Lagrangian multipliers to derive the corresponding dual problem, which is omitted here. Instead, the dual problem $D^{(k)}$ with the Lagrangian multipliers y_i as the design variables is simply given as:

$$\begin{aligned} D^{(k)}: \quad & \text{maximize} \quad W(\underline{y}) \\ & \text{subject to} \quad \underline{y} \geq \underline{0}. \end{aligned} \quad (3.16)$$

The objective function W ($W : \mathbb{R}^m \rightarrow \mathbb{R}$) reads:

$$W(\underline{y}) = r_0 - \underline{y}^T \cdot (\hat{\underline{g}} - \underline{r}) + \sum_{j=1}^n \left(\frac{p_{0j}^{(k)} + \underline{y}^T \underline{p}_j}{U_j^{(k)} - x_j(\underline{y})} + \frac{q_{0j}^{(k)} + \underline{y}^T \underline{q}_j}{x_j(\underline{y}) - L_j^{(k)}} \right). \quad (3.17)$$

Svanberg suggests a Fletcher-Reeves gradient-based optimization algorithm (a conjugate gradient method) to solve the dual problem. Furthermore, he shows that $D^{(k)}$ has a unique solution, from which the solution to the primary subproblem $P^{(k)}$ can

3.5. SVANDBERG'S METHOD OF MOVING ASYMPTOTES

be computed by substituting y_i into the following expression:

$$x_j(y_i) = \frac{L_j \sqrt{p_{0j} + y_i p_{ij}} + U_j \sqrt{q_{0j} + y_i q_{ij}}}{\sqrt{p_{0j} + y_i p_{ij}} + \sqrt{q_{0j} + y_i q_{ij}}}, \quad (3.18)$$

enabling the completion of step 3 in above-mentioned algorithm.

Applying the Method Of Moving Asymptotes for finding suitable parameters in a dynamic stall model, as Mert [10] did, would mean the following:

- The design variables are the three time constants T_V , T_f , T_p and a_{cd} , a correction factor for an assumed vortex drag in the DYNSTALL model:

$$\underline{x} = [T_V, T_f, T_p, a_{cd}]^T. \quad (3.19)$$

- The objective function is the error produced by the dynamic stall model, usually computed by subtracting experimental results from the model output. However, the objective function must be a scalar quantity, so there needs to be a way to reduce errors in the quantities lift, drag and pitching moment to one single quantity. Mert uses two different objective functions, one based solely on the lift:

$$g_0 = \sum_{i=1}^k \left[(c_{L, \text{sim}}(t_i) - c_{L, \text{exp}}(t_i))^2 \right] \quad (3.20)$$

and the other is a weighted sum of error in lift and error in drag:

$$g_0 = \eta \cdot \sum_{i=1}^k \left[(c_{L, \text{sim}}(t_i) - c_{L, \text{exp}}(t_i))^2 \right] + (1 - \eta) \cdot \sum_{i=1}^k \left[(c_{D, \text{sim}}(t_i) - c_{D, \text{exp}}(t_i))^2 \right]. \quad (3.21)$$

The DYNSTALL model only computes lift and drag, but no pitching moment, so the error is determined only from these two properties. However, in a dynamic stall model that provides all three quantities, the error should be determined from all three components as well. This issue and the issue of how the error should be computed will be dealt with in Section 3.4.

- Apart from being non-zero to avoid division by zero, there are no natural, explicit constraints on the time constants. One could argue that a negative time constant would be non-physical, so it makes sense to require strict pos-

3.5. SVANDBERG'S METHOD OF MOVING ASYMPTOTES

itivity as well. There is no upper bound, however, so the constraints on the time constants can be written down as:

$$x_1 = T_V > 0, \quad (3.22a)$$

$$x_2 = T_f > 0, \quad (3.22b)$$

$$x_3 = T_p > 0. \quad (3.22c)$$

There are no natural, implicit constraints on any of the design variables in the form of $g_i \leq 0$.

However, the Method of Moving Average will not converge without implicit constraints, as Mert [10] reported. Only after modifying it into a problem with additional constraints by introducing artificial variables, he could achieve convergence of the algorithm. For every artificial variable z_i added to the objective function (in the form of $d_i z_i + d_i z_i^2$, where $d = \{d \in \mathbb{R} \mid d > 0\}$ is an arbitrary, positive coefficient) in Eq. 3.23, a new constraint is created. With these additional artificial variables and constraints, the modified primary problem then reads:

$$\begin{aligned} \tilde{P}^{(k)}: \quad & \text{minimize} \quad \sum_{j=1}^n \left(\frac{p_{0j}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{0j}^{(k)}}{x_j - L_j^{(k)}} \right) + r_0^{(k)} + \sum_{i=1}^m (d_i z_i + d_i z_i^2) \\ & \text{subject to} \quad \sum_{j=1}^n \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right) + r_i^{(k)} - z_i \leq \hat{g}_i \\ & \quad \quad \quad x_{j,\min} \leq x_j \leq x_{j,\max} \\ & \quad \quad \quad z_i \geq 0. \end{aligned} \quad (3.23)$$

The purpose of these artificial variables is to 'shift' the problem into a feasible region, from where the algorithm can find a solution. In that case, the z_i will become zero in the optimal solution of $\tilde{P}^{(k)}$ and $\tilde{P}^{(k)}$ becomes equal to $P^{(k)}$.

This modification of the Method of Moving Averages will improve it, but not necessarily make it converge to a globally optimal solution in any case, the reason for that being its convexity requirement.

A set $\mathbb{S} \in \mathbb{R}^n$ is convex if for all $\underline{x}_1, \underline{x}_2 \in \mathbb{S}$ and if for all $\lambda = \{\lambda \in \mathbb{R} \mid 0 \leq \lambda \leq 1\}$:

$$\lambda \cdot \underline{x}_1 + [1 - \lambda] \cdot \underline{x}_2 \in \mathbb{S}. \quad (3.24)$$

3.5. SVANDBERG'S METHOD OF MOVING ASYMPTOTES

In words: $\mathbb{S}$ is convex if for all points $\underline{x}_1, \underline{x}_2$ in $\mathbb{S}$, the line connecting $\underline{x}_1$ and $\underline{x}_2$ is also in $\mathbb{S}$.

Let $\mathbb{S}$ be a convex set in $\mathbb{R}^n$. Then, it is possible to define: A function C ($C: \mathbb{S} \rightarrow \mathbb{R}$) is convex if for all $\underline{x}_1, \underline{x}_2 \in \mathbb{S}$ and for all $\lambda = \{\lambda \in \mathbb{R} \mid 0 \leq \lambda \leq 1\}$:

$$C(\lambda \cdot \underline{x}_1 + [1 - \lambda] \cdot \underline{x}_2) \leq \lambda \cdot C(\underline{x}_1) + [1 - \lambda] \cdot C(\underline{x}_2). \quad (3.25)$$

For a one-dimensional function, this means the graph of the function C needs to lie below (or on) the line drawn through the endpoints ($C(x_1)$ and $C(x_2)$) of the interval. The property of convex functions most relevant to optimization is the fact that a minimum is a global minimum and thus, methods that find local minima automatically yield the global minimum when applied to convex problems. Hence, the Method of Moving Averages requires convexity. However, it can be shown that finding optimal parameter for a Beddoes-Leishman dynamic stall model does not satisfy convexity simply by computing the error over a sufficiently large range. This has been done for the airfoil S809 at Reynolds number $Re = 1,000,000$ for all eight test cases. Figures 3.2 and 3.3 show the error ϵ_{total} of the AeroDyn dynamic stall model when the initial value of time constant T_f is varied for cases 106 and 108. The individual error norms e_L , e_D and e_M are plotted as well. The optimal value of T_{f0} according to ϵ_{total} lies around 5 and the Method of Moving Averages will converge to these values if started with initial guesses in the vicinity. It will also successfully find an optimum of T_f if the error would be computed as in Mert's DYNSTALL (Equations (3.20) and (3.21)) or any other linear combination of individual error norms e_L , e_D and e_M . However, if the algorithm should be run from values beyond the peak, e.g. $T_{f0} > 120$, then it will move towards larger and larger values (without upper bound), but not find the global optimum.

Obviously, adding a constraint in form of an upper limit (i.e. $0 < T_{f0} \leq 50$) would restrict the search space to a convex set and make the Method of Moving Average converge. However, knowledge of necessary constraints can not be expected – especially when dealing with new, untested airfoils. So, in general, finding optimal parameter for a Beddoes-Leishman dynamic stall model is not a convex optimization problem and the Method of Moving Averages will not necessarily yield such optimal values.

3.5. SVANDBERG'S METHOD OF MOVING ASYMPTOTES

Figure 3.2: Error over parameter T_{f0} for test case TC106

3.5. SVANDBERG'S METHOD OF MOVING ASYMPTOTES

Figure 3.3: Error over parameter T_{f0} for test case *TC108*

3.6. ALTERNATIVE CHOICES FOR AN OPTIMIZATION METHOD

3.6 Alternative Choices For An Optimization Method

Instead of the Method of Moving Averages, other algorithms suitable for finding global optima need to be considered. There are plenty available – often categorized into deterministic (converging to an optimal solution in finite time), stochastic (converging to an optimal solution in indefinite time) or heuristic (higher-level algorithm to find solutions close to the optimal solution). A desired algorithm should combine several features, most importantly it should guarantee to find an optimal solution, but not get stuck at local optima. It should require no gradients, preferably, because computing gradients can become cumbersome – especially with increasing number of design variables (i.e. $\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \frac{\partial g}{\partial x_3}, \dots$). Last, but not least, an optimization method should be applicable not only in academic, but also in industrial environments and thus, satisfy certain requirements in terms of computational cost, i.e. requirements of computational resources, run-time. A common must-have to fit into the industrial workflow is the capability of running an optimization overnight (from the end of the work day until next morning).

3.6.1 Brute Force Optimization/Grid Search

Probably the simplest optimization algorithm – evaluating all possible solutions and picking the best one – is a *brute force* algorithm. But evaluating literally any potential solution can not be realized, because it would mean an infinite number of samples. Even though the search space is limited to a finite number of points through finite machine precision, this discretized search space still represents a prohibitively large number of evaluations. It needs to be further reduced by adjusting the distance between the sample points. Strictly speaking, this will prevent the algorithm from finding exact optima. However, solutions close to optima can be considered optimal with precision defined by the specified sample distance.

A pseudo code might look like this:

```

set f_opt:= 1,000,000
set number of samples N_samples:= 10,000
set interval [x_lower; x_upper]
set sample width delta_x:= (x_upper-x_lower) / N_samples
for i:= [1:N_samples]
  compute next sample location x_i:= x_lower + i * delta_x
  evaluate f(x_i)
  if (f(x_i) < f_opt) then

```

3.6. ALTERNATIVE CHOICES FOR AN OPTIMIZATION METHOD

```

    x_opt:= x_i
    f_opt:= f(x_i)
  end if
end for

```

Traversing the search space between x_{lower} and x_{upper} with N samples in one dimension will increase the number of samples by a factor N for every design variable added, rendering an optimization program utilizing this technique prohibitive, because it would be too costly in terms of computational resources. Numerous alternative algorithms, significantly faster and more efficient than brute force, exist. It is used here as a reference and most importantly as a tool to find an actual optimal solution, which will then serve as reference to the other methods. With that being said, a brute force algorithm was implemented for the purpose of finding proper optimal parameters and thus, to provide reference values for the other methods.

3.6.2 Random Search/Random Optimization

A very simple and straightforward global optimization method/algorithm is *random optimization* – or *random search*. The search space is probed at randomly selected/picked locations (an analogy would be throwing darts blindfolded/with closed eyes) and the smallest found objective value is returned as optimum. A pseudo code might look like this:

```

set f_opt:= 1,000,000
set number of samples N_samples:= 10,000
set interval [x_lower; x_upper]
for i:= [1:N_samples]
  compute next sample location x_i:= random([x_lower; x_upper])
  evaluate f(x_i)
  if (f(x_i) < f_opt) then
    x_opt:= x_i
    f_opt:= f(x_i)
  end if
end for

```

In older publications [32, 33, 34, 35], the random optimization is not described as taking probes at randomly selected within the search space, but rather as performing consecutive/iteratively steps from the current location in a randomly chosen

3.6. ALTERNATIVE CHOICES FOR AN OPTIMIZATION METHOD

direction and length, which resembles the physical process of Brownian motion, but should nonetheless equivalently traverse the entire domain/search space. A random search method can be applied to virtually any problem, it is straightforward and easy to implement. Moreover, it will find global optima, provided that a sufficient number of samples have been checked. One of the issues with random optimization is that it is not clear how many sample evaluations it requires to arrive at the optimal solution. Similar to the grid search, a solution very close to the optimal solution has to be accepted. For complex problems, where evaluations of the samples are computationally costly, other, more heuristic methods should be preferred over a random search optimization. In this work, a random optimization is employed as a "dumb" algorithm for the purpose of reference.

3.6.3 Evolution Strategies

In computer science, an *Evolution Strategy* (ES) is an optimization technique belonging to the family of *Evolutionary Computation* techniques, characterized by adopting concepts from the field of biology. It is a metaheuristic method – meaning a method that is executed on a simplified problem (in terms of completeness, accuracy, precision, optimality) such that a solution can be found quicker than by applying exact methods, at the price of not finding an analytically exact optimum, but an optimum 'good enough' to be considered exact – with stochastic character, since it involves random processes. A typical algorithm for an evolution strategy can be written as follows:

generate first generation of random samples and evaluate their objective function

repeat until convergence/time limit:

 choose two samples as parents

 generate new individual from selected parents

 mutate new individual and evaluate its objective function

 select best samples for next generation

In an Evolution Strategy, four concepts transferred from evolutionary biology are modelled to form an optimization method. These are:

- The biological process of *Reproduction* by which a new organism is produced from its parent organisms is characterized by propagation of the parents features and qualities to the child organism. In nature, these features and

3.6. ALTERNATIVE CHOICES FOR AN OPTIMIZATION METHOD

qualities are carried by the organism's gene pool, for which the equivalent entities in the optimization process are the design variables. They define an individual sample and according to them, two separate samples can be distinguished. The equivalent step in an Evolution Strategy optimization would be to create a new sample solution from two existing, 'parent' solutions.

- An important concept in the reproduction of organisms is the *Recombination*. The offspring produced during procreation is supposed to be not a mere copy of its parent(s), but it should combine features from both of them, while at the same time being unique. In nature, the genes hosting the qualities of the parent organisms are combined in the so-called chromosome crossover step, forming a new, distinct set of genes for the offspring organism. In the optimization method, this is modelled by combining the parent design variables in some fashion, such that a new, distinct solution is generated, when creating a new solution during the Reproduction step.
- *Mutation* describes a modification of genes due to random, external influences, which can have either beneficial or detrimental effects on the features of the organism. Lactose tolerance is considered an example for the former, a beneficial mutation in the human species that occurred thousands of years ago, whereas an example for the latter is skin cancer caused by ultra-violet light exposure. Mutation can easily be modelled as a random adjustment of the design variables.
- The term *Natural Selection*, coined by Charles Darwin according to his principle of *Survival of the Fittest*, denotes the process by which those genetic features increasing the probability for an organism to survive will spread among succeeding generations. As a simple example, if a rabbit is quicker than others, he may escape the foxes more often than his fellows and produce more offspring, which are then more likely to be quicker as well. Applied to the optimization process, this means that those samples with better objective value will make it into the next generation, while others will be discarded.

Another concept found in Evolution Strategies is that of *Elitism* according to which a certain number of the top-ranked sample solutions will be taken over into the next generation without subjecting it to the Recombination step. This is supposed to ensure that the best solutions attained so far will not be lost during the process.

The ideas of Evolutionary Computation were introduced in the 1950s, which evolved

3.6. ALTERNATIVE CHOICES FOR AN OPTIMIZATION METHOD

into Genetic Algorithms, Evolutionary Algorithms and others. In the 1970s, Rechenberg [36, 37] and Schwefel [38] developed Evolution Strategies. Research in this field has been ongoing and numerous applications have been found.

3.6.4 Implementation Of An Evolution Strategy

The actual implementation of an Evolution Strategy is described as follows: The first generation of $N_{samples}$ sample solutions is initialized from random values for the design variables (parameters and time constants) and their objective value is evaluated by executing the dynamic stall model based on the corresponding parameters and time constants. The next generation is constructed by picking the best N_{elite} samples. According to the elitism concept, the best solutions are supposed to be kept untouched, such that they might not be lost in the reproduction step. Here, 50% of the N_{elite} samples are indeed kept as they are, the other 50% is subjected to a mutation step, in which each of the design variable is allowed to be varied by a random value within the range of 1% of its current value. The purpose of this is to allow some slight modification of the design variables representing a local random search within regions of good prospects. $N_{elite} = 0.20 \cdot N_{samples}$ has turned out a good choice.

Next, a number of N_{new} new, randomly generated solutions are added to each next generation. This will introduce fresh input to the group of samples and allow the search to occur in regions of the search space uncovered so far. The number N_{new} is a parameter to be adjusted. It is recommended to set it rather small with respect to the generation size, e.g. $N_{new} = 0.10 \cdot N_{samples}$. The actual majority of new solutions will be generated from reproduction: picking the two top-ranked solutions from the previous generation as parents, a new solution, i.e. a new set of design variables, is generated by randomly picking either design variable of parent one or parent two, followed by a mutation step. After that, the two remaining top-ranked solutions serve as parents for new samples, until all remaining spots in the next generation are filled. Then, the objective value of all new samples needs to be evaluated.

In the absence of a meaningful stopping criterion, these steps are repeated for a fixed number of iterations/generations (N_{gen}).

3.7. PARALLEL COMPUTING

3.7 Parallel Computing

No matter whether the brute force, random search or the genetic algorithm, large numbers of executions of the dynamic stall program are necessary for all of them. In light of this, speed-up via parallel computation can turn out to be immensely helpful. The dynamic stall routine is a sequential algorithm and hence, not easily parallelized. Implementing it on a vector processor (i.e. executing one instance operating on multiple data sets) would require substantial effort, which does not seem justifiable. Instead, running several, independent instances at the same time – each running on one data set, i.e. the original data partitioned into smaller packages – is possible, which can either be achieved on one computer featuring a *multi-processor* architecture or on a distributed system of computers connected by a network. There are several reasons for the former being the better choice: Besides not requiring machine-to-machine communication, operating on one computer can be implemented on "off the shelves" hardware. Moreover, neither does it require additional software (e.g. MPI) nor an adaption to the dynamic stall program, except for a proper encapsulation.

In order to achieve such a parallelization, the optimization routine is integrated into a simple master-slave framework, in which the master program, i.e. the actual optimization algorithm, runs as one instance on one CPU – just like on a regular program – and whenever an evaluation of the dynamic stall model is required, the corresponding input files are generated and distributed among folders on the hard disk associated to the slaves. The slave program, running as another instance on the same machine, regularly checks its associated folder for newly created subfolders and will begin execution once a new directory with input files has been detected. After completing the evaluation, the results (errors and – optionally – lift, drag, moment data over time) are written to the hard disk, from where the master program can read them. Increased performance can be achieved by running multiple instances of the slave program, such that several executions of the dynamic stall routine can be performed parallel at the same time. So, the process and the integration into the optimization algorithm for the three optimization programs can be described as categorized in Table 3.1. As it can be seen from that table, the slave program performs identical operations and thus, can be used unaltered in all three optimization methods.

The workstations in the computer labs of the School of Mechanical and Aerospace Engineering at NTU feature multi-core processors with 12 CPUs. Using one CPU

3.7. PARALLEL COMPUTING

for the master program and running 10 slave programs on the same machine, will allow a (theoretical) tenfold speed-up and consume 11 out of 12 CPUs leaving one processor for other work. In an industrial setting, the engineer would use $n - 1$ processors out of his n CPU machine for the optimization, allowing him to still use the computer to work on his day-to-day business.

Figure 3.4: Run-time

Figure 3.4 shows the run-times of the three parallelized optimization routines with random search, brute force and evolutionary algorithm using different numbers of processors (ranging from 1 to 10). The sample problem that served as example involved 256 evaluations (250 in case of the evolutionary algorithm) of all eight test cases each. It was executed ten times and the run-times were averaged over these ten runs. The run-times across the three optimization algorithms lie all very close together indicating that the evaluation of the dynamic stall program takes the majority of the total run-time such that the actual algorithm does not play a significant role.

The well-known *Amdahl's law* [39] gives the theoretical performance gain, or speed-up, s when using parallel computing resources. Mathematically, it states:

3.7. PARALLEL COMPUTING

Figure 3.5: Speed-up

$$s = \frac{1}{1 - f_p + \frac{f_p}{n}}. \quad (3.26)$$

f_p stands for the fraction of the program code that will benefit from parallelizing, i.e. that part of the code that will be executed faster when running on multiple processors. If the entire program code will experience a performance increase (in that particular case: $f_p = 1$), a maximum possible speed-up can be achieved of $s = n$, where n is the number of processors. However, in real world settings, this can not be achieved. With more and more processors deployed, the performance gain will rather asymptotically approach an upper threshold prescribed by equation (3.26). This phenomenon manifests itself in a horizontal asymptote in a graph visualizing the speed-up such as Figure 3.5. The decaying slope of the curves in that figure gives evidence for that. If significantly more than 10 processors would have been investigated (which is not possible, as the workstations in the computer labs of the School of Mechanical and Aerospace Engineering at NTU have 12

3.7. PARALLEL COMPUTING

CPUs), the horizontal asymptote could be found at:

$$\lim_{n \rightarrow \infty} \frac{1}{1 - f_p + \frac{f_p}{n}} = \frac{1}{1 - f_p}. \quad (3.27)$$

Table 3.2 gives an overview of the characteristic properties speed-up s of the cases discussed in this Section. From each case, the parallelizable fraction f_p is computed and the values turn all out to close to 1, indicating two things: First, the program code yields excellent speed gains upon parallelization, which is due to the fact that it is essentially a sequential code executed in multiple instances in parallel. Secondly, it indicates it was indeed worthwhile to perform the parallelization.

3.7. PARALLEL COMPUTING

	Master program	Slave program
Random Optimization	The master program sets up all samples and generates a folder with input files containing the corresponding parameter values for each sample. Once all folders have been created, it will begin to check these folders for the creation of result files containing the errors and read them.	The slave programs check for newly created folders in its associated folders. When a new folder has been detected, a sample with the specified parameter values will be executed and the results written into that folder, after which the program will return to checking for newly created folders.
Brute Force Optimization	The master program sets up all samples and generates a folder with input files containing the corresponding parameter values for each sample. Once all folders have been created, it will begin to check these folders for the creation of result files containing the errors and read them. (Identical to Random Optimization)	The slave programs check for newly created folders in its associated folders. When a new folder has been detected, a sample with the specified parameter values will be executed and the results written into that folder, after which the program will return to checking for newly created folders. (Identical to Random Optimization)
Evolutionary Algorithm	The master program sets up the samples for the first generation and generates a folder with input files containing the corresponding parameter values for each sample. Once all folders have been created, it will begin to check these folders for the creation of result files containing the errors and read them. Having read all error files, the optimization algorithm proceeds with selection, reproduction and mutation steps upon which the master program will create folders for the subsequent/following generation of samples, repeating the cycle.	The slave programs check for newly created folders in its associated folders. When a new folder has been detected, a sample with the specified parameter values will be executed and the results written into that folder, after which the program will return to checking for newly created folders. (Identical to Random Optimization)

Table 3.1: Master slave framework

3.7. PARALLEL COMPUTING

n CPUs	brute force		random search		evolutionary alg.	
	s	f_p	s	f_p	s	f_p
2	1.921	0.959	1.920	0.958	1.889	0.941
4	3.403	0.941	3.733	0.962	3.268	0.925
5	3.907	0.930	3.998	0.937	3.855	0.926
8	5.442	0.933	5.641	0.940	4.911	0.910
10	6.045	0.918	6.127	0.920	5.728	0.908

Table 3.2: Overview of speed-up properties

Chapter 4

Optimization Of Time Constants And Parameters In Beddoes-Leishman Dynamic Stall Models

4.1 Optimization Of One Parameter

In order to show the optimization algorithm's ability, an optimization with the genetic algorithm was performed with one parameter, the time constant T_f , as design variable. It was applied to the S809 airfoil at Reynolds number 1,000,000 pitching according to test cases 101 to 108. The corresponding figures are listed in Appendix A.3 beginning on page 197. An optimal solution was found after iterating 10 generations with $T_f = 3.6646$, the default value was $T_f = 3.0$. These values are slightly apart and it shows that the effect on the curves of lift, drag and pitching moment is also very small throughout the eight cases. What can be said is that the increased value of T_f leads to more delay in the response.

4.1.1 Test Case 101

The lift curve of test case 101 has a little more hysteresis towards the end of upstroke and during the entire downstroke. In the drag curves, slightly less hysteresis appears during the second half of upstroke and entire downstroke. The pitching moment curves show slight deviations only during a short segment of upstroke.

4.1. OPTIMIZATION OF ONE PARAMETER

Apart from that, the curves seem to lie on top of each other.

4.1.2 Test Case 102

There is more overshoot at the end of upstroke in the lift. During downstroke, the delayed response leads to an undershoot with respect to the default T_f , which is detrimental at the beginning of downstroke, but beneficial for the rest of the returning motion. The drag shows only minor deviations during both up- and downstroke, the curves seem to mostly lie on top of each other. This is also the case for the pitching moment. The differences appear only in segments of larger gradients.

4.1.3 Test Case 103

The lift response in test case 103 generates more hysteresis during both upstroke and downstroke, but here, only during second half of downstroke. This also appears to be the case for the drag. The delay of the pitching moment response leads to smaller hysteresis during a small fraction of upstroke, most of it is not observable, because the curves lie on top of each other.

4.1.4 Test Case 104

The delayed response increases the integral area under the lift curve during the second half of the upstroke. When the airfoil moves back down, the added time lag of the dashed curve makes the hysteresis negative during the first half, but then positive again during the remaining downward motion. This appears to roughly balance each other out, such that only the upstroke seems relevant. The delayed drag during upstroke increases the area of hysteresis as well, which explains why the optimization algorithm has preferred it. The difference between the two pitching moment curves becomes apparent only in areas of larger slope, otherwise, the curves lie on top of each other. The differences are miniscule, however.

4.1.5 Test Case 105

The results of case 105 resemble those of test case 102 in that the deviation in lift become visible at the end of upstroke, where the overshoot increases the area under the curve. At the beginning of downstroke, the dash curve ($T_f = 3.6646$) seems ahead of the full curve ($T_f = 3.0$), i.e. negative delay. Towards the end of the returning motion, the delay is reestablished, but with an undershoot resulting a

4.1. OPTIMIZATION OF ONE PARAMETER

reduction of the integrated area under the curve. Drag and pitching moment results deviate only slightly from the default curve, namely in those segments with larger gradients.

4.1.6 Test Case 106

In test case 106, differences between the two lift curves become apparent for larger angles only, even though lift gradients towards the lower dead center are bigger. The increased value of T_f has the effect of overshoots towards the end of upstroke and during the first half of the returning motion, both increasing the hysteresis. Drag and pitching moment curves show no apparent deviations from each other.

4.1.7 Test Case 107

The lift curve of case 107 shows a delayed response with a substantial increment in surface area and hysteresis during the second half of the upstroke. Moving back down, only in the second half of downstroke, the delay increases the area under the curve. In the first half of downstroke, the delay does not lead to more hysteresis. This also holds for the drag results: an increase in hysteresis is observed during the second half of up- and downstroke each, but not in other segments of the motion. Overall, the behaviour of lift and drag curves upon modifying T_f in test case 107 resembles that of 104, most likely due to the similar pitching kinematic. Also, the pitching moment lie on top of each other, deviations are only recognizable during segments of larger gradients.

4.1.8 Test Case 108

Compared to the default results, the area under the lift curve is larger when T_f is set to 3.6646, which is due to the increased delay during the middle of upstroke. When the airfoil moves back down, the lift results are slightly larger during the first half (resulting in negative hysteresis) and then change to become smaller than the default values increasing the hysteresis. The drag during upstroke also has this added hysteresis during the middle segment. Deviations in the pitching moment appear during upstroke only during two segments following sharp kinks (second order gradients) in the curves, but not in the middle segment of the upward motion where deviations of the other two loads are found.

4.2. OPTIMIZATION OF SEVERAL PARAMETERS – S809 AIRFOIL

parameter	T_{f0}	T_p	T_{V0}	T_{VL}	A_1	A_2	b_1	b_2
default value	3.0	1.7	5.0	35.0	0.3	0.7	0.14	0.53
optimized value	6.04	0.15	4.67	32.14	1.19	7.38	3.40	5.47

Table 4.1: Overview of design variables (S809)

4.2 Optimization Of Several Parameters – S809 Airfoil

Next, the optimization is applied to the S809 airfoil again at all eight test cases and identical flow conditions (Reynolds number 1,000,000), but this time, all eight parameters (T_f , T_p , T_V , T_{VL} , A_1 , A_2 , b_1 , b_2) are design variables of the optimization. After 50 generations (each generation 100 solutions, 20 kept for elitism), the best sample is considered as optimal solution. Its values are given in table 4.1. All figures with graphical results can be found in Appendix A.4 beginning on page 209. They are discussed in the following sections.

4.2.1 Test Case 101

The optimization of test case 101 has left the lift curve with a smaller lower loop, while the upper loop has expanded. The point of vortex shedding appears slightly earlier ($\approx 0.5^\circ$). The drag and pitching moment curves are both reduced in width rendering them a worse fit than the original curves with default values.

4.2.2 Test Case 102

Test case 103 has benefitted from the optimization. This can be seen from the lift curve, which has less hysteresis in the lower loop, but more hysteresis in the upper loop. Overall, it gives it a better match to the experimental results. This can also be said about the drag: after optimization, the drag curve's shape and its reduced hysteresis match the experimental curve significantly better. Unfortunately, the reduced hysteresis in the pitching moment affects it the opposite way: it is now a worse fit for the experimental results.

4.2.3 Test Case 103

The lift results of test case 103 after optimization are only hardly different from the original solution. During upstroke, the optimized curve is ahead of the default curve shifting the separation point towards a slightly lower angle. The following

4.2. OPTIMIZATION OF SEVERAL PARAMETERS – S809 AIRFOIL

flow regime of vortex shedding and load fluctuations increases the area underneath the curve and reduces the error in lift. On the way back down, the optimized lift curve remains ahead resulting in segments of better and worse results. The drag is visibly affected around 17° to 21° , where it can be observed that the slopes of the c_D curve are reduced during both up- and downward motions. Due to being ahead of the drag curve with default values, the error with respect to experimental data is reduced to a small extent. The pitching moment is similarly affected between 17° to 21° due to running ahead of the solution with default values.

4.2.4 Test Case 104

The lift curve of the optimized lift curve features a huge additional contribution during the second half of the upstroke, covering much more of the experimental results. The minor loss of surface area due to a higher prediction of c_L during downstroke is more than overcompensated by this. Overall, the optimized results are far better than the default. What is interesting is that the crossing up- and downstroke curves has disappeared. This characteristic change could indicate potentially a different reattachment behaviour. The drag results of test case 104 show reduced hysteresis during the second half of upstroke and downstroke each, both with beneficial effects. During the beginning of upstroke, reduced drag values add to this, further improving the prediction. The optimized pitching moment curve experiences a shift towards more negative values around the upper dead center. The area underneath the curves between 15° and 22° is shrunk to a zone from roughly 17° to 20° with seemingly no net improvement.

4.2.5 Test Case 105

The modified parameter set increases the surface area underneath the upper loop of the lift curve, while the lower loop is shrunk. This has the effect of an overall better fit of the experimental lift in test case 105 in terms of the error. The drag is predicted substantially better after optimization as well: reduced c_D during upstroke and increased c_D during returning motion lie much closer to the experimental data. On the other hand, the reduced surface underneath the pitching moment curve has the opposite effect: the optimized data gives worse results.

4.2. OPTIMIZATION OF SEVERAL PARAMETERS – S809 AIRFOIL

4.2.6 Test Case 106

In test case 106, the optimized lift curve features an expanded upper loop, while the lower loop is shrunk. Also, the point of vortex shedding occurs earlier (from above 15° to below 15°), which gives an overall better estimate of lift. The reduced hysteresis of drag fits better to the experimental data. The premature vortex shedding contributes to an error reduction as well, whereas for the pitching moment this is not the case. Here, the shrunk area covered by the c_M curves yields a worse match to the experimental data.

4.2.7 Test Case 107

In the c_L results of test case 107, an outcome very similar to case 104 can be seen: the lift is significantly improved during second part of upstroke by adding additional contribution from about 19.4° onwards, whereas the default curve simply continued to decay. The downstroke is slightly worse, because lift is overpredicted, but it is overcompensated by the mentioned immense gain during upstroke. The drag is also improved, because the bulge towards the end of downwards motion (18° and lower) is capped, which outweighs the worse lift estimate during the second part of upstroke (21° and onwards). The area covered by the c_M curve between 17° and 20° is reduced. Also, the area around the upper dead center is shifted towards more negative values into the area covered by the experimental results. Both contributes to a better pitching moment prediction.

4.2.8 Test Case 108

The optimized lift curve of test case 108 features less delay as can be seen from the characteristic point where vortex shedding is initiated (between 15° and 16° as well as between 19.5° and 20.5°). Earlier triggering of vortex shedding yields less error during upstroke, because more area is covered underneath the c_L curve. However, during the first half of the returning downstroke, the increased lift produces additional error, but that is reversed (or even compensated) during the remainder of the downstroke. Overall, an improved lift prediction is achieved. This is also the case for the drag. Near the lower dead center, the surface area between the curves is reduced, which is beneficial, because it was 'negative' surface area (upstroke lift higher than downstroke lift). The region with the largest area contribution between 16° and 21° gets improved, too. Finally, the optimization has increased the surface

4.3. OPTIMIZATION OF SEVERAL PARAMETERS – S825 AIRFOIL

covered between 21° and 27° , where the the pitching moment results covered the most area, and reduced the area around 15° , where some error was present.

4.3 Optimization Of Several Parameters – S825 Airfoil

As another airfoil, the S825 airfoil is subjected to the optimization process. Again, all eight parameters (T_f , T_p , T_V , T_{VL} , A_1 , A_2 , b_1 , b_2) are design variables. With identical parameters for the optimization routine (50 generations, each generation 100 solutions each generation, 20 elitist solutions), the best sample is considered as optimal solution. Its values are given in table 4.2. All figures with graphical results can be found in Appendix A.5 beginning on page 221.

Due to lack of time, the results will only be briefly discussed. Overall, the results attained from the optimization are in good agreement with the results of the dynamic stall model with default values. That means that the optimization achieved to find parameter values that produce results considered to be optimal. Hence, the optimization can be considered successful. There are a few cases, where the default results could be outperformed. In particular, these are lift in cases 104 or the drag in test case 105. However, better performance in one load often comes with worse performance in the other two loads. In many cases, the optimization improves the load prediction by shrinking the covered area underneath the curves, in particular in those segments where large errors exist. Examples are drag of test cases 102, 106 (Figs. A.269, A.281) and the pitching moment in cases 106 and 108 (Figs. A.282, A.288). Also, the fluctuating loads caused by the vortex shedding are often 'melted away' (Figs. A.271, A.280), manifested by a large value of T_{VL} . It was increased from 11.0 to 58.01.

parameter	T_{f0}	T_p	T_{V0}	T_{VL}	A_1	A_2	b_1	b_2
default value	3.0	1.7	6.0	11.0	0.3	0.7	0.14	0.53
optimized value	2.56	0.03	9.74	60.78	2.92	1.77	4.11	8.30

Table 4.2: Overview of design variables (S825)

4.4. OPTIMIZATION OF SEVERAL PARAMETERS – LS(1)-0147MOD AIRFOIL

4.4 Optimization Of Several Parameters – LS(1)-0147MOD Airfoil

The third airfoil investigated is the NASA LS(1)-0417MOD. The optimization is performed with all eight dynamic stall parameters as parameters (listed with default values and optimized values in table 4.3) and optimization parameters identical to the previous section (50 generations, each generation 100 solutions, 20 kept for elitism). Appendix A.6 lists all figures.

The results of the dynamic stall model after optimization and with default values are again in good agreement with each other, so the optimization has satisfied its main requirement, i.e. to find optimal values (assuming the default values were the best possible). Providing further evidence for this, the values of three time constants (apart from T_{VL}) are rather close to the default values. The lift prediction of test case 104 as well as drag in case 105 are substantially improved, which appears to be universal throughout all three airfoils. Better predictions in one of the loads are also observed to be accompanied by worse performance in the other two loads for the LS(1)-0417MOD airfoil. In many cases, the optimization improves the load prediction by shrinking the covered area underneath the curves, in particular in those segments where large errors exist. The optimization tends to increase T_{VL} a lot across all three airfoils. This reduces or flattens out the fluctuations from the vortex shedding.

parameter	T_{f0}	T_p	T_{V0}	T_{VL}	A_1	A_2	b_1	b_2
default value	5.0	1.7	8.0	15.0	0.3	0.7	0.14	0.53
optimized value	2.49	1.88	9.71	55.82	6.97	0.27	5.32	0.86

Table 4.3: Overview of design variables (LS(1)-0417MOD)

Chapter 5

Conclusions

5.1 Summary And Conclusions

This thesis has taken a closer look at the parameters of a Beddoes-Leishman dynamic stall model. Finding suitable values for these empirical parameters is tedious work and an automated method or algorithm to do that work represents a significant improvement. Using an optimization method to find suitable parameters by minimizing the error between predicted and experimental loads is a solution to this issue. Previous researchers employed a convex optimization method (Method of Moving Averages), but it could be shown in this work that such method will generally not succeed. Instead, a global optimization method is required, for which Evolution Strategy was implemented. Applying such an optimization method in conjunction with a Beddoes-Leishman dynamic stall model led to parameter sets for three different airfoils (S809, S825, LS(1)-0417MOD) yielding overall good results in predicting loads with the dynamic stall model.

The following conclusions can be drawn:

- Overall, the modification of the full set of eight parameters and time constants does not fundamentally change the outcome of the load prediction, i.e. the characteristic shape of lift, drag or pitching moment curves, but rather magnitudes, time delay or vortex shedding frequencies of the responses. That means the fundamental character of the dynamic stall model is not overthrown – as it was not intended. Instead, modifying the empirical parameters via an optimization algorithm can serve as a tool for finding a set of best choice parameters or for fine-tuning.
- The optimized results are pretty close to those based on default values, so

5.2. PUBLICATIONS

optimization can be said to at least find what 'human trial&error' can do.

- Improvements by the optimization process can not only be found in one of the three loads, but throughout all of them, indicating that the optimization is indeed not biased towards one of the quantities.
- Since the exact time signal data of the experimental data was not available, it had to be derived from experimental data depending on angle of attack, not attack. Not only does the conversion from angle of attack into time introduce uncertainty, but also there is uncertainty involved in the process of digitizing the data from plots and graphs.

5.2 Publications

Reinecke, A., Gupta, A. and Chua, L.P., *Finding Optimal Parameters For A Beddoes-Leishman Dynamic Stall Model*, Wind Energy, In Preparation

Appendix A

Appendix

A.1 Figures – Performance Of NREL Dynamic Stall Model

A.1.1 Test Case 101

Figure A.1: lift coefficient for test case 101

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

Figure A.2: drag coefficient for test case 101

Figure A.3: pitching moment coefficient for test case 101

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

A.1.2 Test Case 102

Figure A.4: lift coefficient for test case 102

Figure A.5: drag coefficient for test case 102

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

Figure A.6: pitching moment coefficient for test case 102

A.1.3 Test Case 103

Figure A.7: lift coefficient for test case 103

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

Figure A.8: drag coefficient for test case 103

Figure A.9: pitching moment coefficient for test case 103

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

A.1.4 Test Case 104

Figure A.10: lift coefficient for test case 104

Figure A.11: drag coefficient for test case 104

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

Figure A.12: pitching moment coefficient for test case 104

A.1.5 Test Case 105

Figure A.13: lift coefficient for test case 105

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

Figure A.14: drag coefficient for test case 105

Figure A.15: pitching moment coefficient for test case 105

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

A.1.6 Test Case 106

Figure A.16: lift coefficient for test case 106

Figure A.17: drag coefficient for test case 106

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

Figure A.18: pitching moment coefficient for test case 106

A.1.7 Test Case 107

Figure A.19: lift coefficient for test case 107

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

Figure A.20: drag coefficient for test case 107

Figure A.21: pitching moment coefficient for test case 107

A.1. FIGURES – PERFORMANCE OF NREL DYNAMIC STALL MODEL

A.1.8 Test Case 108

Figure A.22: lift coefficient for test case 108

Figure A.23: drag coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.24: pitching moment coefficient for test case 108

A.2 Figures – Sensitivity Of The Dynamic Stall Model

A.2.1 Parameter A_1

Test Case 101

Figure A.25: lift coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.26: drag coefficient for test case 101

Figure A.27: pitching moment coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 102

Figure A.28: lift coefficient for test case 102

Figure A.29: drag coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.30: pitching moment coefficient for test case 102

Test Case 103

Figure A.31: lift coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.32: drag coefficient for test case 103

Figure A.33: pitching moment coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 104

Figure A.34: lift coefficient for test case 104

Figure A.35: drag coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.36: pitching moment coefficient for test case 104

Test Case 105

Figure A.37: lift coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.38: drag coefficient for test case 105

Figure A.39: pitching moment coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 106

Figure A.40: lift coefficient for test case 106

Figure A.41: drag coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.42: pitching moment coefficient for test case 106

Test Case 107

Figure A.43: lift coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.44: drag coefficient for test case 107

Figure A.45: pitching moment coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 108

Figure A.46: lift coefficient for test case 108

Figure A.47: drag coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.48: pitching moment coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

A.2.2 Parameter A_2

Test Case 101

Figure A.49: lift coefficient for test case 101

Figure A.50: drag coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.51: pitching moment coefficient for test case 101

Test Case 102

Figure A.52: lift coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.53: drag coefficient for test case 102

Figure A.54: pitching moment coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 103

Figure A.55: lift coefficient for test case 103

Figure A.56: drag coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.57: pitching moment coefficient for test case 103

Test Case 104

Figure A.58: lift coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.59: drag coefficient for test case 104

Figure A.60: pitching moment coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 105

Figure A.61: lift coefficient for test case 105

Figure A.62: drag coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.63: pitching moment coefficient for test case 105

Test Case 106

Figure A.64: lift coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.65: drag coefficient for test case 106

Figure A.66: pitching moment coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 107

Figure A.67: lift coefficient for test case 107

Figure A.68: drag coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.69: pitching moment coefficient for test case 107

Test Case 108

Figure A.70: lift coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.71: drag coefficient for test case 108

Figure A.72: pitching moment coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

A.2.3 Parameter b_1

Test Case 101

Figure A.73: lift coefficient for test case 101

Figure A.74: drag coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.75: pitching moment coefficient for test case 101

Test Case 102

Figure A.76: lift coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.77: drag coefficient for test case 102

Figure A.78: pitching moment coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 103

Figure A.79: lift coefficient for test case 103

Figure A.80: drag coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.81: pitching moment coefficient for test case 103

Test Case 104

Figure A.82: lift coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.83: drag coefficient for test case 104

Figure A.84: pitching moment coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 105

Figure A.85: lift coefficient for test case 105

Figure A.86: drag coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.87: pitching moment coefficient for test case 105

Test Case 106

Figure A.88: lift coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.89: drag coefficient for test case 106

Figure A.90: pitching moment coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 107

Figure A.91: lift coefficient for test case 107

Figure A.92: drag coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.93: pitching moment coefficient for test case 107

Test Case 108

Figure A.94: lift coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.95: drag coefficient for test case 108

Figure A.96: pitching moment coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

A.2.4 Parameter b_2

Test Case 101

Figure A.97: lift coefficient for test case 101

Figure A.98: drag coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.99: pitching moment coefficient for test case 101

Test Case 102

Figure A.100: lift coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.101: drag coefficient for test case 102

Figure A.102: pitching moment coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 103

Figure A.103: lift coefficient for test case 103

Figure A.104: drag coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.105: pitching moment coefficient for test case 103

Test Case 104

Figure A.106: lift coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.107: drag coefficient for test case 104

Figure A.108: pitching moment coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 105

Figure A.109: lift coefficient for test case 105

Figure A.110: drag coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.111: pitching moment coefficient for test case 105

Test Case 106

Figure A.112: lift coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.113: drag coefficient for test case 106

Figure A.114: pitching moment coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 107

Figure A.115: lift coefficient for test case 107

Figure A.116: drag coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.117: pitching moment coefficient for test case 107

Test Case 108

Figure A.118: lift coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.119: drag coefficient for test case 108

Figure A.120: pitching moment coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

A.2.5 Time Constant T_f

Test Case 101

Figure A.121: lift coefficient for test case 101

Figure A.122: drag coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.123: pitching moment coefficient for test case 101

Test Case 102

Figure A.124: lift coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.125: drag coefficient for test case 102

Figure A.126: pitching moment coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 103

Figure A.127: lift coefficient for test case 103

Figure A.128: drag coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.129: pitching moment coefficient for test case 103

Test Case 104

Figure A.130: lift coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.131: drag coefficient for test case 104

Figure A.132: pitching moment coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 105

Figure A.133: lift coefficient for test case 105

Figure A.134: drag coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.135: pitching moment coefficient for test case 105

Test Case 106

Figure A.136: lift coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.137: drag coefficient for test case 106

Figure A.138: pitching moment coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 107

Figure A.139: lift coefficient for test case 107

Figure A.140: drag coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.141: pitching moment coefficient for test case 107

Test Case 108

Figure A.142: lift coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.143: drag coefficient for test case 108

Figure A.144: pitching moment coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

A.2.6 Time Constant T_p

Test Case 101

Figure A.145: lift coefficient for test case 101

Figure A.146: drag coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.147: pitching moment coefficient for test case 101

Test Case 102

Figure A.148: lift coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.149: drag coefficient for test case 102

Figure A.150: pitching moment coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 103

Figure A.151: lift coefficient for test case 103

Figure A.152: drag coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.153: pitching moment coefficient for test case 103

Test Case 104

Figure A.154: lift coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.155: drag coefficient for test case 104

Figure A.156: pitching moment coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 105

Figure A.157: lift coefficient for test case 105

Figure A.158: drag coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.159: pitching moment coefficient for test case 105

Test Case 106

Figure A.160: lift coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.161: drag coefficient for test case 106

Figure A.162: pitching moment coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 107

Figure A.163: lift coefficient for test case 107

Figure A.164: drag coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.165: pitching moment coefficient for test case 107

Test Case 108

Figure A.166: lift coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.167: drag coefficient for test case 108

Figure A.168: pitching moment coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

A.2.7 Time Constant T_V

Test Case 101

Figure A.169: lift coefficient for test case 101

Figure A.170: drag coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.171: pitching moment coefficient for test case 101

Test Case 102

Figure A.172: lift coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.173: drag coefficient for test case 102

Figure A.174: pitching moment coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 103

Figure A.175: lift coefficient for test case 103

Figure A.176: drag coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.177: pitching moment coefficient for test case 103

Test Case 104

Figure A.178: lift coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.179: drag coefficient for test case 104

Figure A.180: pitching moment coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 105

Figure A.181: lift coefficient for test case 105

Figure A.182: drag coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.183: pitching moment coefficient for test case 105

Test Case 106

Figure A.184: lift coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.185: drag coefficient for test case 106

Figure A.186: pitching moment coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 107

Figure A.187: lift coefficient for test case 107

Figure A.188: drag coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.189: pitching moment coefficient for test case 107

Test Case 108

Figure A.190: lift coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.191: drag coefficient for test case 108

Figure A.192: pitching moment coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

A.2.8 Time Constant T_{VL}

Test Case 101

Figure A.193: lift coefficient for test case 101

Figure A.194: drag coefficient for test case 101

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.195: pitching moment coefficient for test case 101

Test Case 102

Figure A.196: lift coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.197: drag coefficient for test case 102

Figure A.198: pitching moment coefficient for test case 102

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 103

Figure A.199: lift coefficient for test case 103

Figure A.200: drag coefficient for test case 103

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.201: pitching moment coefficient for test case 103

Test Case 104

Figure A.202: lift coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.203: drag coefficient for test case 104

Figure A.204: pitching moment coefficient for test case 104

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 105

Figure A.205: lift coefficient for test case 105

Figure A.206: drag coefficient for test case 105

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.207: pitching moment coefficient for test case 105

Test Case 106

Figure A.208: lift coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.209: drag coefficient for test case 106

Figure A.210: pitching moment coefficient for test case 106

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Test Case 107

Figure A.211: lift coefficient for test case 107

Figure A.212: drag coefficient for test case 107

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.213: pitching moment coefficient for test case 107

Test Case 108

Figure A.214: lift coefficient for test case 108

A.2. FIGURES – SENSITIVITY OF THE DYNAMIC STALL MODEL

Figure A.215: drag coefficient for test case 108

Figure A.216: pitching moment coefficient for test case 108

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

A.3 Figures – Optimization Of One Parameter – S809 Airfoil

A.3.1 Test Case 101

Figure A.217: lift coefficient for test case 101

Figure A.218: drag coefficient for test case 101

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

Figure A.219: pitching moment coefficient for test case 101

A.3.2 Test Case 102

Figure A.220: lift coefficient for test case 102

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

Figure A.221: drag coefficient for test case 102

Figure A.222: pitching moment coefficient for test case 102

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

A.3.3 Test Case 103

Figure A.223: lift coefficient for test case 103

Figure A.224: drag coefficient for test case 103

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

Figure A.225: pitching moment coefficient for test case 103

A.3.4 Test Case 104

Figure A.226: lift coefficient for test case 104

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

Figure A.227: drag coefficient for test case 104

Figure A.228: pitching moment coefficient for test case 104

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

A.3.5 Test Case 105

Figure A.229: lift coefficient for test case 105

Figure A.230: drag coefficient for test case 105

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

Figure A.231: pitching moment coefficient for test case 105

A.3.6 Test Case 106

Figure A.232: lift coefficient for test case 106

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

Figure A.233: drag coefficient for test case 106

Figure A.234: pitching moment coefficient for test case 106

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

A.3.7 Test Case 107

Figure A.235: lift coefficient for test case 107

Figure A.236: drag coefficient for test case 107

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

Figure A.237: pitching moment coefficient for test case 107

A.3.8 Test Case 108

Figure A.238: lift coefficient for test case 108

A.3. FIGURES – OPTIMIZATION OF ONE PARAMETER – S809 AIRFOIL

Figure A.239: drag coefficient for test case 108

Figure A.240: pitching moment coefficient for test case 108

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

A.4 Figures – Optimization Of Multiple Parameters – S809 Airfoil

A.4.1 Test Case 101

Figure A.241: lift coefficient for test case 101

Figure A.242: drag coefficient for test case 101

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

Figure A.243: pitching moment coefficient for test case 101

A.4.2 Test Case 102

Figure A.244: lift coefficient for test case 102

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

Figure A.245: drag coefficient for test case 102

Figure A.246: pitching moment coefficient for test case 102

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

A.4.3 Test Case 103

Figure A.247: lift coefficient for test case 103

Figure A.248: drag coefficient for test case 103

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

Figure A.249: pitching moment coefficient for test case 103

A.4.4 Test Case 104

Figure A.250: lift coefficient for test case 104

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

Figure A.251: drag coefficient for test case 104

Figure A.252: pitching moment coefficient for test case 104

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

A.4.5 Test Case 105

Figure A.253: lift coefficient for test case 105

Figure A.254: drag coefficient for test case 105

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

Figure A.255: pitching moment coefficient for test case 105

A.4.6 Test Case 106

Figure A.256: lift coefficient for test case 106

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

Figure A.257: drag coefficient for test case 106

Figure A.258: pitching moment coefficient for test case 106

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

A.4.7 Test Case 107

Figure A.259: lift coefficient for test case 107

Figure A.260: drag coefficient for test case 107

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

Figure A.261: pitching moment coefficient for test case 107

A.4.8 Test Case 108

Figure A.262: lift coefficient for test case 108

A.4. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S809 AIRFOIL

Figure A.263: drag coefficient for test case 108

Figure A.264: pitching moment coefficient for test case 108

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

A.5 Figures – Optimization Of Multiple Parameters – S825 Airfoil

A.5.1 Test Case 101

Figure A.265: lift coefficient for test case 101

Figure A.266: drag coefficient for test case 101

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

Figure A.267: pitching moment coefficient for test case 101

A.5.2 Test Case 102

Figure A.268: lift coefficient for test case 102

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

Figure A.269: drag coefficient for test case 102

Figure A.270: pitching moment coefficient for test case 102

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

A.5.3 Test Case 103

Figure A.271: lift coefficient for test case 103

Figure A.272: drag coefficient for test case 103

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

Figure A.273: pitching moment coefficient for test case 103

A.5.4 Test Case 104

Figure A.274: lift coefficient for test case 104

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

Figure A.275: drag coefficient for test case 104

Figure A.276: pitching moment coefficient for test case 104

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

A.5.5 Test Case 105

Figure A.277: lift coefficient for test case 105

Figure A.278: drag coefficient for test case 105

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

Figure A.279: pitching moment coefficient for test case 105

A.5.6 Test Case 106

Figure A.280: lift coefficient for test case 106

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

Figure A.281: drag coefficient for test case 106

Figure A.282: pitching moment coefficient for test case 106

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

A.5.7 Test Case 107

Figure A.283: lift coefficient for test case 107

Figure A.284: drag coefficient for test case 107

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

Figure A.285: pitching moment coefficient for test case 107

A.5.8 Test Case 108

Figure A.286: lift coefficient for test case 108

A.5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – S825 AIRFOIL

Figure A.287: drag coefficient for test case 108

Figure A.288: pitching moment coefficient for test case 108

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

A.6 Figures – Optimization Of Multiple Parameters – LS(1)- 417MOD Airfoil

A.6.1 Test Case 101

Figure A.289: lift coefficient for test case 101

Figure A.290: drag coefficient for test case 101

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

Figure A.291: pitching moment coefficient for test case 101

A.6.2 Test Case 102

Figure A.292: lift coefficient for test case 102

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

Figure A.293: drag coefficient for test case 102

Figure A.294: pitching moment coefficient for test case 102

5. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – (1)-417MOD AIRFOIL

6.3 Test Case 103

Figure A.295: lift coefficient for test case 103

Figure A.296: drag coefficient for test case 103

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

Figure A.297: pitching moment coefficient for test case 103

A.6.4 Test Case 104

Figure A.298: lift coefficient for test case 104

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

Figure A.299: drag coefficient for test case 104

Figure A.300: pitching moment coefficient for test case 104

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

A.6.5 Test Case 105

Figure A.301: lift coefficient for test case 105

Figure A.302: drag coefficient for test case 105

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

Figure A.303: pitching moment coefficient for test case 105

A.6.6 Test Case 106

Figure A.304: lift coefficient for test case 106

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

Figure A.305: drag coefficient for test case 106

Figure A.306: pitching moment coefficient for test case 106

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

A.6.7 Test Case 107

Figure A.307: lift coefficient for test case 107

Figure A.308: drag coefficient for test case 107

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

Figure A.309: pitching moment coefficient for test case 107

A.6.8 Test Case 108

Figure A.310: lift coefficient for test case 108

A.6. FIGURES – OPTIMIZATION OF MULTIPLE PARAMETERS – LS(1)-417MOD AIRFOIL

Figure A.311: drag coefficient for test case 108

Figure A.312: pitching moment coefficient for test case 108

Bibliography

- [1] Gross, D. and Harris, F., "Prediction of in-flight stalled airloads from oscillating airfoil data," *25th Annual Forum of the American Helicopter Society*, Washington, DC, 14-16 May 1969.
- [2] Gormont, R., "A mathematical model of unsteady aerodynamics and radial flow for application to helicopter rotors," Tech. rep., Boeing Vertol Company, May 1973.
- [3] Leishman, J. and Beddoes, T., "A generalised model for airfoil unsteady aerodynamic behaviour and dynamic stall using the indicial method," *Proceedings of the 42nd Annual forum of the American Helicopter Society*, Washington, DC, June 1986, pp. 243–266.
- [4] Leishman, J. and Beddoes, T., "A Semi-Empirical Model for Dynamic Stall," *Journal of the American Helicopter society*, Vol. 34, No. 3, 1989, pp. 3–17.
- [5] Gupta, S. and Leishman, J., "Dynamic stall modelling of the S809 aerofoil and comparison with experiments," *Wind Energy*, Vol. 9, No. 6, 2006, pp. 521–547.
- [6] Viterna, L. and Jonetzke, D., "Theoretical and experimental power from large horizontal-axis wind turbines," Technical Report NASA-TM-82944, National Aeronautics and Space Administration, Cleveland, OH (USA). Lewis Research Center, 1982.
- [7] Du, Z. and Selig, M., "A 3-D stall-delay model for horizontal axis wind turbine performance prediction," *1998 ASME Wind Energy Symposium, Aerospace Sciences Meetings*, Reno, NV, USA, January 1998, pp. 9–19.
- [8] Corrigan, J. J. and Schillings, J. J., "Empirical model for stall delay due to rotation," *Proceedings of the American Helicopter Society Aeromechanics Specialists Conference*, San Francisco, CA, USA, January 1994.
- [9] Pierce, K. and Hansen, A., "Prediction of Wind Turbine Rotor Loads Using the Beddoes-Leishman Model for Dynamic Stall," *Journal of Solar Energy Engineering*, Vol. 117, No. 3, August 1995, pp. 200–204.

BIBLIOGRAPHY

- 0] Mert, M., "Optimization of semi-empirical parameters in the FFA-Beddoes dynamic stall model," Tech. Rep. FFA TN-1999-37, Flygtekniska Försöksanstalten (FFA), June 1999.
- 1] Svanberg, K., "The method of moving asymptotes—a new method for structural optimization," *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 2, 1987, pp. 359–373.
- 2] Björck, A., "DYNSTALL: subroutine package with a dynamic stall model," Tech. Rep. FFAP-V-110, Flygtekniska Försöksanstalten (FFA), June 2000.
- 3] González, A., "DYSTOOL Stability Aerodynamic Tool for 2D Airfoil Based on the Beddoes-Leishman Model," Tech. Rep. IN-523/069-en, Wind Energy Department - Fundación CENER-CIEMAT, Avenida Ciudad de la Innovación, 7 - 31621-Sarriguren (Navarra) Spain: CENER, 2014.
- 4] González, A. and Mundate, X., "Unsteady modelling of the oscillating S809 aerofoil and NREL phase VI parked blade using the Beddoes-Leishman dynamic stall model," *Journal of Physics: Conference Series*, Vol. 75, No. 1, 2007, pp. 012020.
- 5] "AeroDyn," <https://nwtc.nrel.gov/AeroDyn>, last modified: 25.May.2017.
- 6] "FAST," <https://nwtc.nrel.gov/FAST>, last modified: 25.May.2017.
- 7] Bisplinghoff, R., Ashley, H., and Halfman, R., *Aeroelasticity*, Addison-Wesley, Reading, MA, 1955.
- 8] Leishman, G., "Validation of approximate indicial aerodynamic functions for two-dimensional subsonic flow," *Journal of aircraft*, Vol. 25, No. 10, 1988, pp. 914–922.
- 9] Beddoes, T., "Practical computation of unsteady lift," *Eighth European Rotorcraft Forum*, Aix-En-Provence, France, August 1982.
- 10] Damiani, R. and Hayman, G., *The Dynamic Stall Module for FAST 8*, NREL, April 2016.
- 11] Evans, W. and Mort, K., "Analysis of computed flow parameters for a set of sudden stalls in low-speed two-dimensional flow," Nasa technical note, National Aeronautics and Space Administration, Ames Research Center, Moffett Field, CA, USA, August 1959.
- 12] Beddoes, T., "Onset of leading edge separation effects under dynamic conditions and low Mach number," *Proceedings of the 34th Annual Forum of the American Helicopter Society*, Vol. 15, Washington, DC, USA, 1978.
- 13] Beddoes, T., "Representation of airfoil behaviour," *Vertica*, Vol. 7, No. 2, 1983, pp. 183–197.

BIBLIOGRAPHY

- [24] Ramsay, R., Hoffmann, M., and Gregorek, G., "Effects of grit roughness and pitch oscillations on the S809 airfoil," Airfoil Performance Report NREL/TP-442-7817, The Ohio State University, December 1995.
- [25] Ramsay, R., Janiszewska, J., and Gregorek, G., "Effects of grit roughness and pitch oscillations on the S825 airfoil," Airfoil performance report, The Ohio State University, October 1998.
- [26] Janiszewska, J., Ramsay, R., Hoffmann, M., and Gregorek, G., "Effects of grit roughness and pitch oscillations on the LS(1)-0417MOD airfoil," Airfil Performance Report NREL/TP-442-7819, The Ohio State University, January 1996, revised December 1999.
- [27] Reus, R., Hoffmann, M., and Gregorek, G., "Effects of surface roughness and vortex generators on the LS(1)-0417MOD airfoil," Report NREL/TP-442-6674, National Renewable Energy Laboratory (NREL), December 1995.
- [28] Sheng, W., Galbraith, R. A. M., and Coton, F. N., "On The S809 Airfoil's Unsteady Aerodynamic Characteristics," *Journal of Wind Energy*, Vol. 12, No. 8, 2009, pp. 752–767.
- [29] Butterfield, C., Scott, G., and Musial, W., "Comparison of Wind Tunnel Airfoil Performance Data With Wind Turbine Blade Data," *Journal of Solar Energy Engineering*, Vol. 114, No. 2, 1992, pp. 119–124.
- [30] Somers, D., "Design and Experimental Results for the S809 Airfoil," Report NREL/SR-440-6918, National Renewable Energy Laboratory (NREL), March 1997.
- [31] Leishman, J., *Principles of Helicopter Aerodynamics*, Cambridge aerospace series, Cambridge University Press, 2006.
- [32] Anderson, R., "Recent Advances in Finding Best Operating Conditions," *Journal of the American Statistical Association*, Vol. 48, No. 264, 1953, pp. 789–798.
- [33] Rastrigin, L., "About Convergence of Random Search Method in Extremal Control of Multi-Parameter Systems," *Avtomat. i Telemekh.*, Vol. 24, No. 11, 1963, pp. 1467–1473.
- [34] Karnopp, D., "Random search techniques for optimization problems," *Automatica*, Vol. 1, No. 2, 1963, pp. 111–121.
- [35] Schumer, M. and Steiglitz, K., "Adaptive step size random search," *IEEE Transactions on Automatic Control*, Vol. 13, No. 3, Jun 1968, pp. 270–276.
- [36] Rechenberg, I., *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (Evolution Strategy - Optimization of*

BIBLIOGRAPHY

- technical systems according to principles of biological evolution*), Phd thesis, TU Berlin, 1973.
- [37] Rechenberg, I., *Evolutionsstrategie '94 (Evolution Strategy)*, Frommann-Holzboog, 1994.
- [38] Schwefel, H., *Evolutionsstrategie und numerische Optimierung (Evolutionary strategy and numerical optimization)*, Phd thesis, TU Berlin, 1975.
- [39] Amdahl, G., "Validity of the single processor approach to achieving large scale computing capabilities," *Proceedings of the April 18-20, 1967, spring joint computer conference, AFIPS '67 (Spring)*, American Federation of Information Processing Societies, ACM, Atlantic City, New Jersey, April 1967, pp. 483–485.