



Royal Institute of Technology (KTH)
Aeronautical and Vehicle Engineering
Aerodynamics



Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Standardized Geometry Formats for Aircraft Conceptual Design and Physics-based Aerodynamics and Structural Analyses

Student Research Project of Liana Cöllen

Stockholm, September 2011

First examiner: Prof. Dr.-Ing. Arthur Rizzi
Second examiner: Prof. Dr.-Ing. Volker Gollnick

Affidavit

I, LIANA CÖLLEN (student of mechanical engineering at Hamburg University of Technology, matriculation number 20624406), hereby declare that the following student research project has been written only by the undersigned and without any assistance from third parties. Furthermore, I confirm that no sources have been used in the preparation of this project other than those indicated in the thesis itself. As well as that, the thesis has not yet been submitted, neither in this nor in similar form at any other examination board.

Stockholm, 22. September 2011

Task Description for Liana Cöllen

Matr. Nr. _____

Standardized Geometry Formats for Aircraft Conceptual and Physics-based Aerodynamics and Structural Analyses

Introduction

The German Aerospace Center's (DLR) department of Air Transport Systems and the Royal Institute of Technology's (KTH) department of Aeronautical and Vehicle Engineering are both practicing research into multidisciplinary engineering environments for preliminary aircraft design. Aiming at a more efficient collaboration, a unified data format is intended to be established as a bridge between the two design systems. The two partners selected the DLR-developed Common Parametric Aircraft Configuration Scheme (CPACS) as a basis technology.

Task

In the frame of this thesis a connection from CPACS to KTH's CEASIOM framework and in particular to its components AC Builder, AMB, and SUMO has to be developed. The work is founded on conventional transport aircraft as use case with the Airbus A320 as reference configuration. Aircraft data shall comprise both reference data of real aircraft and novel aircraft designed using the conceptual aircraft design tool VAMPzero. Analyzes with the individual software components shall be conducted and input and output correlations shall be documented in a N2 Chart.

The final step is to perform design studies which make use of components of both design systems. DLR's conceptual design tool VAMPzero shall be coupled via CPACS to KTH's CEASIOM components AMB, Propulsion, and SDSA for a detailed analysis of aerodynamic and flight mechanic behavior. A major aspect of this is the application of different aerodynamics methods which are available in the CEASIOM system (DATCOM, TORNADO, and EDGE). The results of the different methods have to be compared with at least the reference configuration and one varied design in order to permit quantifying the methods' impact on design sensitivities.



Working steps

The main working steps of this thesis are listed in the following points:

- Create an interface between CPACS and CEASIOM
- Set up a N2 Chart of used CEASIOM components
- Conduct design studies with conceptual tools and compare resulting weight, CG & I
- Set up analysis environment with coupled CPACS and CEASIOM tools

Conduct variable fidelity design studies and evaluate the impact of the different aerodynamic methods.

Issue Date: _____

Handover date: _____

Prof. Dr.-Ing. V. Gollnick

Liana Cöllen

Prof. Dr.-Ing. A. Rizzi

Dipl.-Ing. T. Pfeiffer

Contents

List of Figures	vii
List of Tables	ix
Nomenclature	x
Formula symbols	x
Greek formula symbols	xi
Indices	xii
Acronyms	xiv
1 Introduction	1
1.1 Thesis breakdown	2
2 Concept	3
2.1 CPACS	3
2.1.1 Module structure	5
2.1.2 Design environment	6
2.2 VAMPzero	7
2.3 CEASIOM	8
2.3.1 AC Builder	10
2.3.2 SUMO	11
2.3.3 AMB	12
2.3.4 Propulsion	14
2.3.5 SDSA	15
3 CPACS to CEASIOM	18
3.1 CPACS to AC Builder	18
3.1.1 Taper Ratio	20
3.1.2 Airfoils	21
3.1.3 Control surfaces	21
3.1.4 Unconventional aircraft configuration	24
3.2 CPACS to AMB	26
3.3 CPACS to SUMO	26

4	Analysis of Airbus A320	28
4.1	Airbus A320	28
4.2	Input of VAMPzero	29
4.3	Comparison of different geometries	29
4.3.1	Fuselage	32
4.3.2	Wing	33
4.3.3	Horizontal tail	35
4.3.4	Vertical tail	35
4.3.5	Engine	35
4.4	Weight breakdown	37
4.4.1	Structure Weight	40
4.4.2	System Weight	40
4.4.3	Powerplant Weight	42
4.4.4	Operating Items Weight	42
4.4.5	Operational Empty Weight	43
4.4.6	Maximum Zero-Fuel Weight	43
4.4.7	Maximum Takeoff Weight	43
4.4.8	Conclusion	44
4.5	Moments of inertia	44
4.6	Center of gravity	45
4.7	Results of AC Builder	46
4.8	Results of SUMO	47
4.9	Results of AMB	49
4.9.1	Conclusion	54
4.10	Results of Propulsion	55
4.11	Results of SDSA	55
5	N2 Chart	58
6	Conclusion and Suggestion for Future Projects	60
A	Wrapper	62
B	AC Builder	65
C	Weights and Balances	67
D	AMB	69
	Bibliography	72

List of Figures

1.1	Three phases of aircraft design	1
2.1	CPACS Wrapper [1]	3
2.2	Flow-chart of the analysis	4
2.3	Number of interfaces [2]	4
2.4	CPACS structure	5
2.5	Approach to multi-fidelity [2]	7
2.6	Dependencies of the wing span	8
2.7	Convergence history of parameters in VAMPzero	9
2.8	CEASIOM framework [3]	10
2.9	Tree structure of aircraft geometrical data [4]	11
2.10	GUI of the revised AC Builder	12
2.11	AMB CFD architecture [5]	13
2.12	Adaptable fidelity modules [6]	14
2.13	Multi fidelity CFD tools	15
2.14	SDSA structure and functionality [5]	16
2.15	Aerodynamic database of SDSA [7]	17
3.1	Overview of generated files	18
3.2	CPACS to AC Builder	19
3.3	Wing planform geometry	21
3.4	Airfoil definition in CPACS	22
3.5	Imported airfoil into CEASIOM	22
3.6	Control surfaces [8]	23
3.7	Geometric definition in CPACS [8]	23
3.8	Aileron definition in CPACS	24
3.9	Input box to remove segments	25
3.10	Blended Wing Body	25
3.11	Input box to add sections	26
3.12	CPACS to SUMO	27
3.13	Engine definition in SUMO	27
4.1	Drawing of A320 [9]	29
4.2	Computation of wingspan	30

4.3	Comparison of geometries	31
4.4	Thrust at takeoff for one engine	36
4.5	Results of the stand-alone tool <i>Weights and Balances</i>	38
4.6	Result of the Howe method	38
4.7	Weight breakdown of a medium-range aircraft	39
4.8	Extract of the analysis chain	40
4.9	Cöllen model	46
4.10	Modification of the vertical location of the vertical tail	47
4.11	Skeleton of the fuselage, side view	47
4.12	Skeleton of the fuselage, top view	48
4.13	SUMO rendering of the A320	48
4.14	Surface mesh	49
4.15	Volume mesh, refinement 4	50
4.16	GEO TORNADO	51
4.17	Composition of the drag coefficient	52
4.18	Aerodynamic coefficients, Mach = 0.6, $\beta = 0^\circ$	53
4.19	Propulsion output	55
4.20	Trim results	56
4.21	Stability criteria	56
5.1	Composition of N2 Chart [10]	58
5.2	N2 Chart [h=hardcoded, m>manual inserted]	59
B.1	Data of flap, aileron and slat	65
B.2	Elevator data	66
B.3	Rudder data	66
C.1	Weight breakdown of <i>Weights and Balances</i>	68
C.2	Centers of gravity	68
D.1	Reference data	69
D.2	States	69
D.3	Aerodynamic table	70
D.4	EDGE Euler convergence history, Mach = 0.6, $\alpha = 15^\circ, \beta = 0^\circ$	71

List of Tables

3.1	Definition of control surfaces	24
4.1	Airbus A320 specifications [11]	28
4.2	Input parameters of the fuselage	32
4.3	Input parameters of the wing * defined in VAMPzero input file	33
4.4	Parameters of the horizontal tail * defined in VAMPzero input file	35
4.5	Parameters of the vertical tail * defined in VAMPzero input file	36
4.6	Parameters of the engine * defined in VAMPzero input file	37
4.7	Nomenclature of different models	39
4.8	Nomenclature of different methods	39
4.9	Weight breakdown of A320 [kg] * defined in VAMPzero/AC Builder input file	41
4.10	Moments of inertia [kgm^2]	44
4.11	Centers of gravity [m]	45
4.12	Surface mesh data	49
4.13	Volume mesh data	50
4.14	Deviation of aerodynamic coefficients at 1500 iterations	52
4.15	Calculation period per test case	54
A.1	Necessary parameters for AC Builder	62
A.2	Calculated parameters in AC Builder	62
A.3	Input parameters of VAMPzero	64

Nomenclature

Formula symbols

Latin formula symbols

Symbol	Unit	Description
b	[m]	span
c	[m]	local chord
d	[m]	diameter; distance
e	[–]	Oswald efficiency number
g	[m/s ²]	gravity acceleration
h	[m]	maximum height
k	[–]	coefficient of proportionality
l	[m]	length
n	[–]	number of modules
v	[m/s]	velocity
w	[m]	maximum width
y	[m]	arbitrary spanwise location
AR	[–]	Aspect Ratio
C	[–]	aerodynamic coefficient
D	[–]	damping ratio
$Mach$	[–]	Mach number
N	[–]	number of seats abreast
P	[Nm]	roll moment
PAX	[–]	design number of passengers
Q	[Nm]	pitch moment
R	[Nm]	yaw moment
V	[m ³]	volume
W	[kg]	weight

Greek formula symbols

Symbol	Unit	Description
α	[°]	angle of attack
β	[°]	side slip angle
δ	[°]	deflection angle
η	[m]	positioning along the wingspan
λ	[–]	taper ratio
ξ	[m]	positioning along the wing chord
ω	[1/s]	angular frequency
ω_0	[1/s]	eigen angular frequency
Λ	[–]	aspect ratio
Π	[–]	linear factor
Φ	[–]	impulse function

Indices

Index	Description
a	= aileron; aerodynamic
abs	= absolute coordinate system; absolute
apu	= Auxiliary Power Unit
avn	= avionic system
bpax	= carrying passenger number for business aircraft
cab	= cabin
cpax	= carrying passenger number for commercial transport
comp	= interior completion including paint
cons	= consumables and other provisions
decr	= fuel decrement
e	= elevator
ecab	= equivalent cabin
ecs	= Enviromental Control System
elec	= electrical system
eng	= engine
f	= body-fixed
fatt	= cabin attendants
fcnt	= flight control system
fcrew	= flight crew
fe	= viscous drag
furn	= furniture
fus	= fuselage
fusu	= unusable fuel
hyd	= hydraulic system
inst	= instruments
l	= rolling moment
lg	= landing gear
loc	= local coordinate system
misc	= miscellaneous
n	= yawing moment
nac	= nacelle
oper	= operational items
p	= placement
pax	= passengers
pay	= maximum payload
pow	= powerplant
prop	= propeller

Indices

pyl	=	pylon
r	=	rudder
ref	=	reference
sfuel	=	fuel system
stru	=	structure
sys	=	systems
wet	=	wetted
wlet	=	winglet
ATM	=	advanced technology multiplier
D	=	drag
Di	=	induced lift drag
D0	=	zero lift drag
L	=	lift
LE	=	leading edge
M	=	pitching moment
R	=	root
S	=	area
SW	=	sweep angle
TE	=	trailing edge
W	=	weight
X	=	x-axis of the body-fixed coordinate system
Y	=	side force
YEIS	=	year of entry into service of the aircraft design being
Z	=	z-axis of the body-fixed coordinate system

Acronyms

Acr.	Description
uID	= unique Identifier
AMB	= Aerodynamic Model Builder
APU	= Auxiliary Power Unit
ASCII	= American Standard Code for Information Interchange
BWB	= Blended Wing Body
CAD	= Computer-Aided Design
CEASIOM	= Computerized Environment for Aircraft Synthesis and Integrated Optimization Methods
CFD	= Computational Fluid Dynamics
CGNS	= CFD General Notation System
CPACS	= Common Parametric Aircraft Configuration Scheme
DATCOM	= Data Compendium
DLR	= German Aerospace Center
DOF	= Degrees of Freedom
ECS	= Enviromental Control System
FAR	= Federal Aviation Regulations
FCS	= Flight Control System
FCSDT	= Flight Control System Design Module
GUI	= Graphical User Interface
High-Fi	= Higher Fidelity
ICAO	= International Civil Aviation Organization
IGES	= Initial Graphics Exchange Specification
JAR	= Joint Aviation Requirements
KTH	= Kungliga Tekniska Högskolan
LQR	= Linear Quadratic Regulator
MAC	= Mean Aerodynamic Chord
MIL	= Military Handbook of Aviation
MTOW	= Maximum Takeoff Weight
MZFW	= Maximum Zero Fuel Weight
NeoCASS	= Next generation Conceptual Aero Structural Sizing
NACA	= National Advisory Committee for Aeronautics
OEW	= Operational Empty Weight
PASS	= Program for Aircraft Synthesis Studies
RANS	= Reynolds-Averaged Navier-Stokes
SimSAC	= Simulating Aircraft Stability and Control Characteristics for use in Conceptual Design

Acronyms

SAS	=	Stability Augmentation System
SC	=	Stability and Control
SDSA	=	Simulation and Dynamic Stability Analyzer
STREP	=	Specific Targeted Research Project
SUMO	=	Surface Modeler
TetGen	=	Tetrahedral Mesh Generator
VLM	=	Vortex Lattice Method
XML	=	Extensible Markup Language

1 Introduction

Present trends in aircraft design towards configurations with augmented stability or operations in expanded flight envelopes require accurate predictions of the flight-dynamic behavior. The process of aircraft design is split up into three phases depicted in figure 1.1. In the conceptual design phase the configuration arrangement, size, weight, and performance of the aircraft are defined by fulfilling requirements of the customer. Hence, conceptual design is a fluid process where new ideas and problems occur. During the preliminary design phase specialists in the areas of structure, landing gear, and control systems design and analyze the components of the aircraft. Preliminary design has to establish confidence to build the aircraft on time with the estimated costs. In the detail design phase the precise geometry of the components are designed for fabrication considering producibility. Besides accomplishing structure test flights, control systems are simulated and tested by pilots. After successful completion of the three phases in aircraft design the aircraft will be manufactured [12]. Conceptual design is the primary focus of this research project.



Figure 1.1: Three phases of aircraft design

The ability to predict stability and control behavior of an aircraft requires complete and accurate aerodynamic data. Conventionally, forces and moments are provided by wind tunnel measurements of the flight envelope of a wind tunnel model which becomes available in the late design process. Most engineering tools in aircraft design are based upon handbook methods and linear fluid mechanics assumptions. As long as the aircraft remains within the flight envelope these methods constitute reliable data. Nevertheless, knowledge about non-linear flight-dynamic behavior is required to predict augmented-stability aircrafts and aircrafts in expanded flight envelopes. For that reason, CFD (Computational Fluid Dynamics) methods are applied substituting expensive experimental measurements [13].

The department of Aeronautical and Vehicle Engineering at Royal Institute of Technology (KTH) is practicing research in multidisciplinary engineering environments for conceptual aircraft design. By means of the conceptual design tool CEASIOM tables of aerodynamic forces and moments are generated. Additionally, implemented higher fidelity modules based on CFD methods allow analysis of stability and control behavior of the aircraft in the early design process. Currently, decentralized frameworks in the area of multidisciplinary aircraft analysis and optimization are state of the art. Although a multitude of commercial and open source software products are available, the implementation of an optimization framework necessitates

significant efforts [1]. As a first step, this thesis presents a collaboration between the design system CEASIOM and VAMPzero developed at the German Aerospace Center (DLR). Through cooperation of KTH and DLR a converter between the design systems is created using the DLR-developed Common Parametric Aircraft Configuration Scheme (CPACS) as a basis technology.

Within the framework of this thesis a wrapper between CPACS and CEASIOM is developed. The work is founded on a conventional transport aircraft with the Airbus A320 as reference configuration. Subsequently, a performing design study is accomplished with the conceptual design tool VAMPzero coupled via CPACS to CEASIOM. At first resulting masses, moments of inertia and center of gravity of the design tools are compared. This is followed by an analysis of aerodynamic and flight mechanic behavior in the environment of CEASIOM using the components AMB, Propulsion and SDSA. By applying the aerodynamic methods DATCOM, TORNADO, and EDGE Euler the results are evaluated due to the impact on design sensitivities.

1.1 Thesis breakdown

In chapter 2 an overview about the intention of this thesis is given. Furthermore, the motivation and the concept of the standardized geometry format CPACS is introduced. In particular the conceptual design tools VAMPzero and CEASIOM are described. Subsequently, in chapter 3 a wrapper is set up to translate data from CPACS to CEASIOM. In a first step the approach to develop a wrapper between CPACS and AC Builder is presented. Then, the wrapper is extended for data transfer from CPACS to AMB and SUMO. After application of the wrapper the generated Airbus A320 data are analyzed in chapter 4. Firstly, the characteristics of the A320 are described followed by specifications of the input file of VAMPzero. Afterwards, different shapes of the A320 are illustrated and identified based on different data sources. Moreover, the prediction of weights, moments of inertia, and center of gravity of different estimation methods are presented. Within the frame of CEASIOM the results of AC Builder, SUMO, AMB, Propulsion, and SDSA are demonstrated. Subsequently, in chapter 5 the input and output correlations of the individual software components are documented in a N2 Chart. Concluding, the achievements of the thesis are summarized in chapter 6 and suggestions for future projects are made.

2 Concept

The aim of this thesis is to set up a wrapper between the standard syntax definition format CPACS and the conceptual design tool CEASIOM. The wrapper comprises a script, which is able to read the CPACS format and to translate it into the input format of a certain method. After applying this method an output file is generated. The output file is read by the wrapper and is translated back to the format of CPACS as depicted in figure 2.1. With the junction of wrappers to different methods, communication and collaboration via CPACS is feasible. For that reason, the development of wrappers and the extension of the tool-portfolio is a valuable investigation [1].

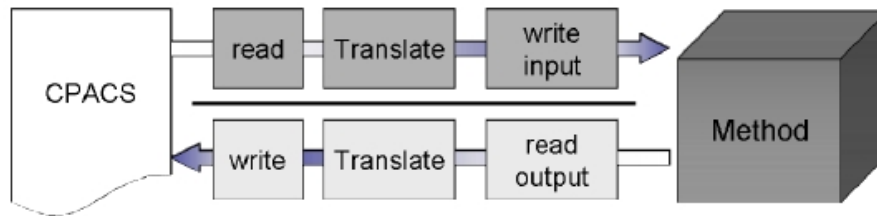


Figure 2.1: CPACS Wrapper [1]

In this thesis a wrapper is developed in order to read data in CPACS format and to translate them into CEASIOM format. The reverse direction of the wrapper is not implemented in this work. As an application of this interface between CPACS and CEASIOM, a reference aircraft configuration in CPACS provided by VAMPzero is used for aerodynamic and flight mechanic analysis in CEASIOM. The sequence of the analyses with the components AC Builder, SUMO, AMB, Propulsion, and SDSA is presented in a flow-chart illustrated in figure 2.2. This thesis is founded on conventional transport aircrafts with the Airbus A320 as a reference configuration. The Airbus A320-family is, besides the similar Boeing-737-family, one of the most sold medium-range aircraft worldwide. Currently, Airbus develops the A320neo, a successor of the A320 including benefits of the A320-family [14]. For that reason, the data of the Airbus A320 are of special interest and associated analyzes are accomplished in this thesis.

To achieve a better comprehension of the software environment CPACS and the conceptual design, the respective tools are described in the following sections. Subsequently, the body of the wrapper is presented in chapter 3.

2.1 CPACS

Expertise and capabilities to analyze aircraft design are distributed over the boundaries of institutions and locations. Bringing together the data of various disciplines provided by valuable design tools, such as VAMPzero and CEASIOM, is a difficult and demanding process.

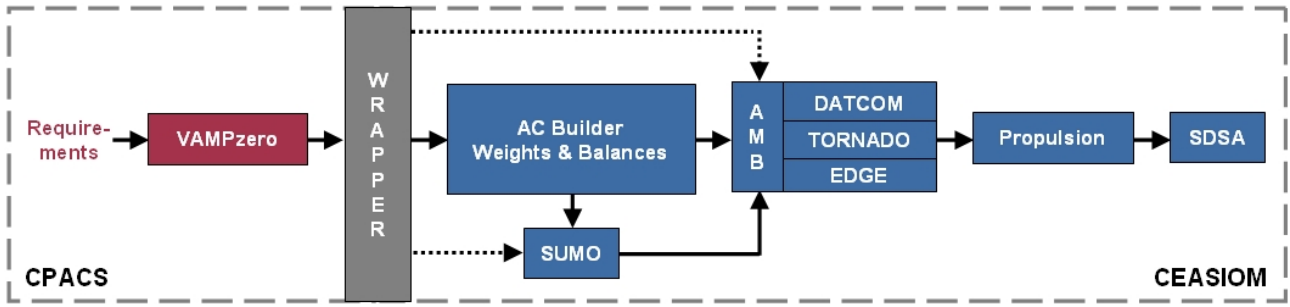


Figure 2.2: Flow-chart of the analysis

Therefore, the design environments have to define a common namespace and exploit design codes adopted to their own code. Design teams in various institutions benefit not only from standardized product descriptions but also from identical coupling of analysis modules, that process information of analysis chains can be exchanged easily.

Regarding efficient data exchange, the critical factor for flexibility of design environments is the number of interfaces. As displayed in figure 2.3, the structure of the connection of analysis modules has a considerable influence on the number of interfaces. The central information model in figure 2.3a holds in contrast to the model in figure 2.3b a reduced number of interfaces, thus the analysis modules get more independent and modifications in one module do not implicitly affect other tools. Based on the usage of the common namespace, the central information model represents the meta-model for all derived analysis models.

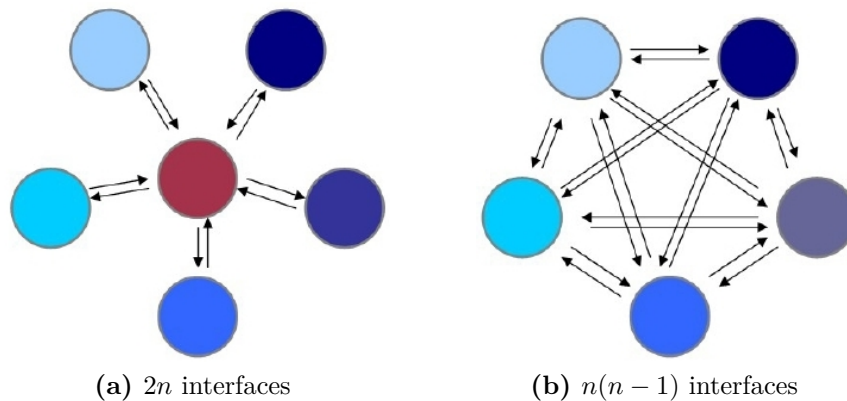


Figure 2.3: Number of interfaces [2]

Since 2005 the DLR has been developing the Common Parametric Aircraft Configuration Scheme (CPACS), a standard syntax definition for the exchange of information within preliminary aircraft design containing detailed type definitions of the air transportation system. Geometric elements such as aircraft, wings, sections, elements, profiles, points, and transformations are defined covering the bandwidth of the named projects. Furthermore, elements such as airports, airline fleets, and flight plans are included [2].

In the following sections the CPACS module structure is introduced and the integration into the design environment is described.

2.1.1 Module structure

CPACS is based on a schema file (XML Schema Definition XSD) containing detailed type definitions of the air transportation system. As an example figure 2.4 shows an extract of the CPACS structure of an A320 configuration. Each element is able to imply further elements, so a complex structure can be described. Moreover, a linkage between the elements with the unique Identifier (uID) of the elements is feasible. The application of the uIDs is supported by TIXI libraries, which guarantees satisfactory handling of CPACS data.

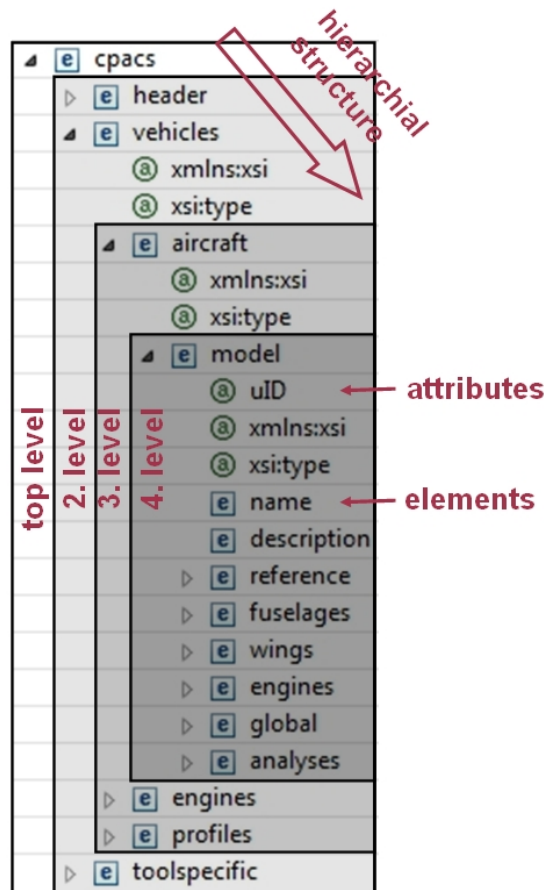


Figure 2.4: CPACS structure

Following the top level of the structure, the second level includes the *header* implying general information about the CPACS data set. Therein information about the data structure name, creator, timestamp of compilation, and version number are stored. Moreover, modification on the data set are documented. The component *vehicles* comprises geometric and performance data of aircrafts in the third level. Since the configuration of the A320 describes a single *model*, it is defined in the fourth level by

- **uID** aircraft uID,
- **name** aircraft name,
- **description** description of the aircraft,

- **reference** reference length, area, and reference point of the aircraft,
- **fuselages** containing the fuselages of the aircraft,
- **wings** containing the wings of the aircraft,
- **engines** containing the position of the engines,
- **global** containing information about the aerodynamic performance map, passenger seats, landing weight, and Mach number at cruise point,
- **analysis** containing the weight breakdown of the aircraft.

The element *engines* includes further information about the engine type comprising geometric data, thrust, and bypass ratio. The shape of the components are appointed in *profiles*. This element is subdivided in *fuselageProfiles* and *wingAirfoils*, which are described by a point list normed within the range of zero up to one [15]. CPACS does not only hold information regarding certain objects but also data of coupled analysis modules. The *toolspecific* data contain information to initiate different options in the analysis modules. Consequently, the analysis modules are able to interact with each other in a progressional way and the framework allows splitting and merging of CPACS data sets [2].

The level depth of the CPACS structure is variable and dependent on the level of detail considered in the specific study. Hence, CPACS is able to define aircrafts with an arbitrary level of fidelity and complexity [1]. Detailed information about the definition of geometric components are available on the website (<http://software.dlr.de/p/cpacs/home>) and are provided in the CPACS documentation, which will be published with an exemplary CPACS file by DLR in a few months.

2.1.2 Design environment

In this section a framework for design environment is introduced consisting of CPACS and a conceptual design module. A multitude of higher fidelity analysis modules including aerodynamic, primary structure, mission analysis, and climate impact are already connected to CPACS. These analysis modules are coupled in a framework depicted in figure 2.5. The multi-fidelity design loop is initialized by top-level requirements containing sufficient geometric information for the first calculations. The results of the design loop should be merged to generate inputs for further iterations and convergence control. For that reason, the DLR developed VAMPzero, a conceptual design module specified in section 2.2. This module generates a CPACS file including, among others, geometry data, weight breakdown, and toolspecific inputs for analysis in higher fidelity (High-Fi) modules. By reducing the granularity of High-Fi results to the level of the conceptual design tool the feedback loop is closed. So the results of the analysis modules are fed back to VAMPzero [2].

Currently, CPACS is in operational use at aeronautical institutes of DLR and has been extended for civil and military aircrafts, rotorcrafts, jet engines, and air transportations systems. CPACS offers the following advantages:

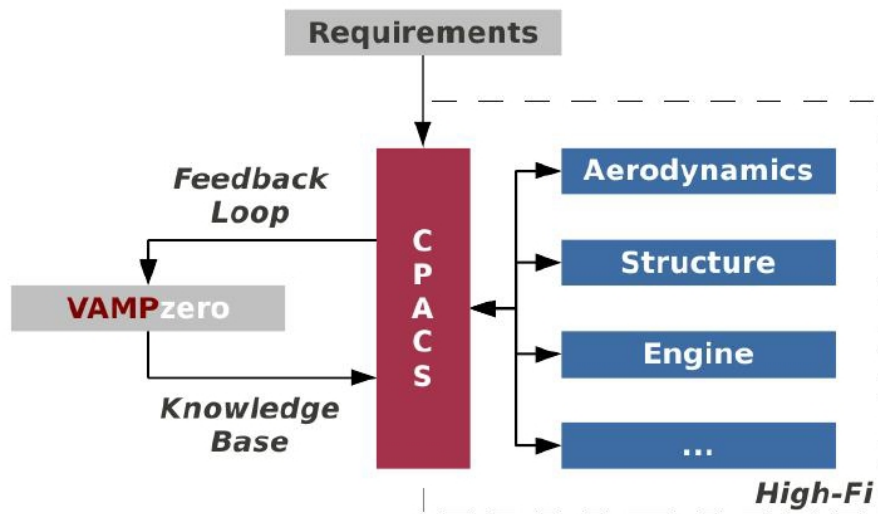


Figure 2.5: Approach to multi-fidelity [2]

- high accessibility,
- various coupled tools,
- comparably small,
- task-specific, and
- supporting libraries

It is desirable to connect further design tools to CPACS to exploit a wider data set. In order to integrate CPACS as a standardized geometry format into aircraft conceptual design, it will be published in the next months.

2.2 VAMPzero

Closing the gap between the input of component-based analysis modules and the design loop requires the usage of conceptual methods. For this purpose the DLR developed the conceptual design module VAMPzero with a main focus on multi-fidelity. VAMPzero is written by Daniel Böhnke and provides requirements for the design process with the claim to be modular, reusable, readable and extensible. The software is able to be run stand-alone.

The new conceptual design code holds an object-oriented structure to increase the modularity and to distinguish aspects such as file handling and convergence control from design aspects such as parameter definitions and calculation methods. So the code is able to be extended and adapted easily. The main classes of the structure are *components* and *parameters* containing general routines, which are inherited or overwritten by specific classes. *Components* are structured in a hierarchical way and part associations with other *components*. Each *component* includes several *parameters* grouped by *disciplines*. While the calculation methods in *components* and *disciplines* are applied for process control, the *parameter* calculation method

includes empirical and analytical knowledge. Additional information about the application of the calculations methods are described in the paper of Böhnke within the framework of the IEEE Aerospace Conference [2].

While the calculation methods access the values of other parameters, a mind-mapping software visualizes the references of the model to support the transparency of the analysis modules. As an example figure 2.6 illustrates the dependencies of the parameter in the center. On the left hand side parameters accessing the value of the central parameter are depicted whereas the parameters on the right hand side are required for the calculation method of the parameter in the center.

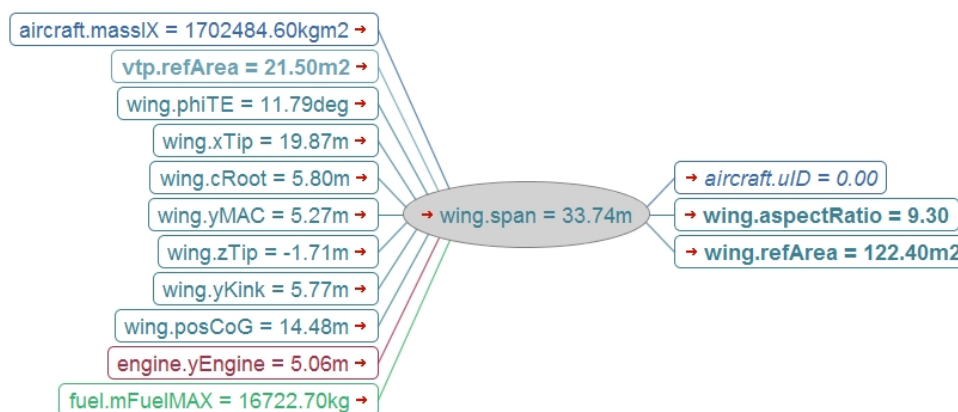


Figure 2.6: Dependencies of the wing span

Before running VAMPzero parameters have to be assigned as a status of being initialized, calculated or fixed. While recently constructed objects are set to initialized, non-fixed parameters will be calculated by one of its calculation methods. Fixing a multitude of parameters might cause conflicts in the calculation process and cause warnings. During the multidisciplinary design processes a few key values, such as maximum takeoff weight, operation empty weight and static thrust, are verified for convergence. VAMPzero reviews the convergence for all parameters separately as displayed in figure 2.7. The relative change of the parameter magnitude is shown against the number of iterations. The depicted convergence history is recorded during the analysis with a minimum number of input values and shows the numerical stability of the code. In order to obtain a stable behavior of the system, a multitude of inputs are needed. So the holistic calculation process closes the design loop by importing CPACS data, calculating the results until convergence is reached, and finally by exporting the results back to CPACS [2].

2.3 CEASIOM

The Computerized Environment for Aircraft Synthesis and Integrated Optimization Methods (CEASIOM) is a framework that integrates discipline-specific tools for conceptual design of aircrafts. The CEASIOM code is developed within the frame of the SimSAC (Simulating Aircraft Stability And Control Characteristics for Use in Conceptual Design) Specific Targeted Research

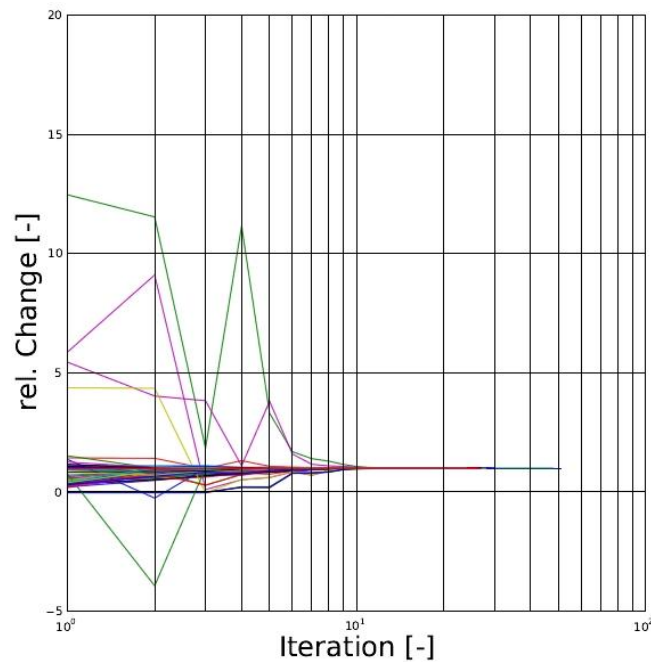


Figure 2.7: Convergence history of parameters in VAMPzero

Project (STREP) promoted by the European Commission's sixth Framework Programme on Research, Technological Development and Demonstration. The software development began in November 2006 and continues to date.

CEASIOM is intended to support engineers in the conceptual design process of aircrafts by generating stability and control data for conceptual design using methods of varying fidelity [5]. With adaptive-fidelity CFD the aerodynamic data set is created to analyze flying qualities using stability and control analysis tools. CEASIOM is a framework tool containing Computer-Aided Design (CAD) and mesh generation tools, CFD codes, and Stability and Control (SC) software for the purpose of aircraft conceptual design. Significant components of CEASIOM are [16]:

1. Geometry module AC Builder:
Customized geometry construction system that allows the user to define the geometry in a XML file by a small number of parameters
2. Aerodynamic module AMB-CFD:
Replacement of current handbook aerodynamic methods with adaptable fidelity CFD modules referred to Tier I (TORNADO and EDGE CFD) and Tier II (RANS)
3. Stability and Control module SDSA:
Simulation and dynamic stability and control analyzer and flying-quality assessor. Flight tests with six Degrees of Freedom (DOF) flight simulation, and performance prediction, also includes human pilot model, Stability Augmentation System (SAS), and a Linear Quadratic Regulator (LQR) based flight control system (FCS) package are among the major functionalities of this module

4. Aeroelastic module NeoCASS:
Quasi-analytical structural analysis methods that support aeroelastic problem formulation and solution
5. Flight Control System design module FCSDT:
Designer toolkit for flight control law formulation, simulation, and technical decision support, permitting flight control system, design philosophy and architecture to be coupled in the early phase of conceptual design

Figure 2.8 presents the functionality, process and dataflow of the CEASIOM software. Modules that are actively involved in this thesis are described in detail in the following sections.

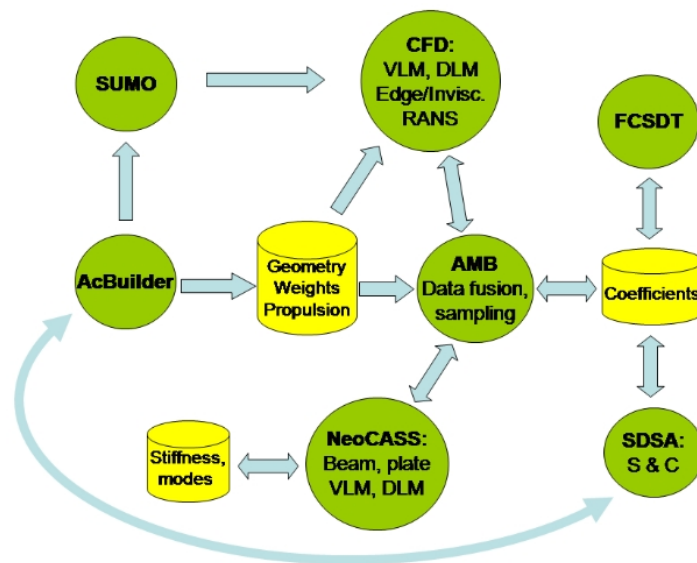


Figure 2.8: CEASIOM framework [3]

The major dataflow between the CEASIOM modules is transferred by a XML file. Including necessary information about the description of the aircraft, the data volume extends within the conceptual design process. The data are arranged in a tree structure, which contains first level elements such as geometric components, weights and balance information. As an example, figure 2.9 shows the structure of the geometrical data. The aircraft possesses child elements consisting of aircraft components, for instance fuselage, wing, horizontal tail, vertical tail, or engines. Each child element includes child elements by themselves. For example, the wing has child elements containing information about the span, area, and control surfaces. In the next level of the tree, the structure of the control surfaces are described.

2.3.1 AC Builder

AC Builder allows the user to visualize the aircraft in the early design process before conducting further computations. The tool is divided into the three modules *Geometry*, *Weights and Balances*, and *Technology*. The module *Geometry* defines the components of the aircraft and

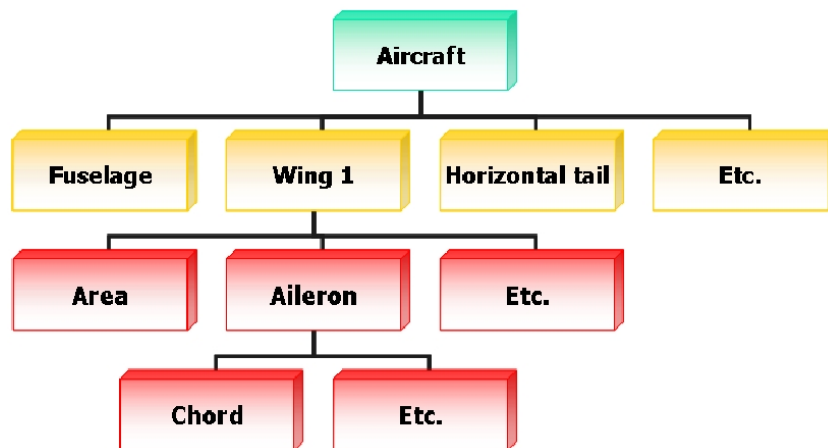


Figure 2.9: Tree structure of aircraft geometrical data [4]

describes the fuel tank and the wingbox. Moreover, an output geometry is generated containing the reference wings of the main wing, horizontal tail, and vertical tail. These information are required to estimate weights, centers of gravity, and the matrix of inertia of the aircraft by the following module *Weights and Balances*. The implemented calculation methods are described in the doctoral thesis of Askin T. Isikveren [17]. The accuracy of the prediction of weights, centers of gravity, and moments of inertia influences the results of the stability and control analysis of the aircraft significantly. The module *Technology* visualizes the input parameters for NeoCASS, where the user is able to define the beam model, aero panels, and spar locations. Furthermore, material characteristics are determined and structural concepts are configured.

AC Builder permits illustrations of conventional aircraft configurations, but the visualization of unconventional aircrafts is restricted due to a limited number of elements (1 fuselage, 2 main wings, 1 horizontal tail, ect.). This led to the development of a revised AC Builder, wherein the user is allowed to add an arbitrary number of lifting surfaces and aircraft components. However, it has to be taken into account that this feature is not adapted in further CEASIOM tools yet. Figure 2.10 shows the Graphical User Interface (GUI) of the revised AC Builder. The left window is implemented in Matlab and allows modification of parameters describing aircraft elements. The window on the right hand side renders the aircraft based on a Java code [18].

2.3.2 SUMO

SUMO is a graphical tool for aircraft geometries and automatic surface mesh generation [19]. The Surface Modeler (SUMO) provides three-dimensional, watertight aircraft geometries satisfactorily for high-fidelity analysis using CFD methods [16]. Hence, SUMO aspires rapid surface modeling of aircraft configurations. SUMO is based on a C++ library for geometric primitives such as b-spline surfaces [19].

In a first step a triangular mesh is generated on the aircraft surface. The triangular elements are based on a three-dimensional in-sphere criterion. In comparison to the Delaunay methods

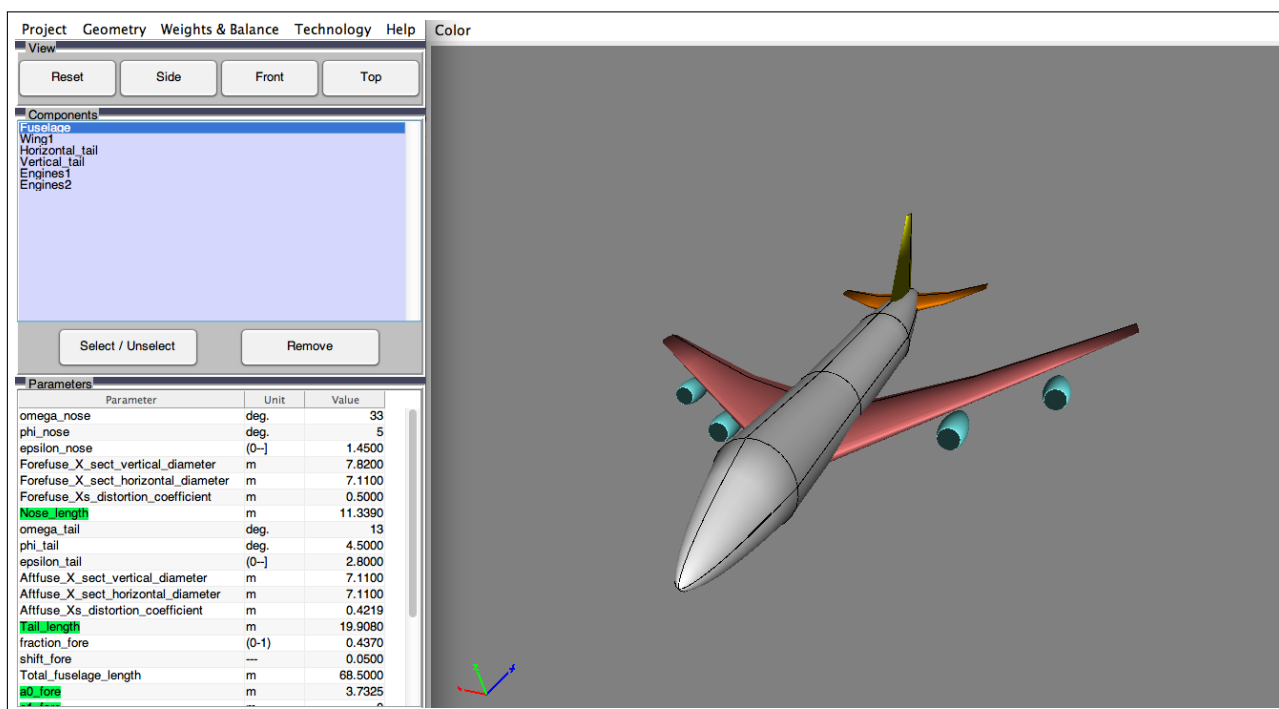


Figure 2.10: GUI of the revised AC Builder

they yield a better mesh quality for skewed surfaces such as thin, swept delta wings. Moreover, triangular elements generates surface meshes easier, at least when a sufficient quantity of elements are used [19]. Based on surface meshes, unstructured volume meshes are created by the Tetrahedral Mesh Generator (TetGen) [20]. Thereby volume meshes are generated using tetrahedral elements to fill the space between the aircraft and the farfield [21]. Continuous geometry data are able to export as IGES format whereas volume meshes are exported using CGNS, a bmesh file for the CFD solver EDGE [22]. Additionally, volume meshes are able to be saved as TetGens plain ASCII format [16].

2.3.3 AMB

The Aerodynamic Model Builder (AMB) is a component of CEASIOM, which generates tables of aerodynamic forces and moments for flight dynamics analysis. So CFD is able to describe non-linear flight dynamic behavior of the aircraft. Required data are created by sampling and data fusion from diverse sources [16]. AMB provides four methods to estimate aerodynamic forces and moments. The information flow of the prediction methods is visualized in figure 2.11. Thereby, data sources have to be taken into consideration for each method. Applied methods in the aircraft design process of this thesis are described below. The method based on Reynolds-Averaged Navier-Stokes (RANS) equations is not exploited in this work.

DATCOM

Data Compendium (DATCOM) is a handbook implying methods determining stability and control characteristics for different aircraft configurations. DATCOM is a semi-empirical method

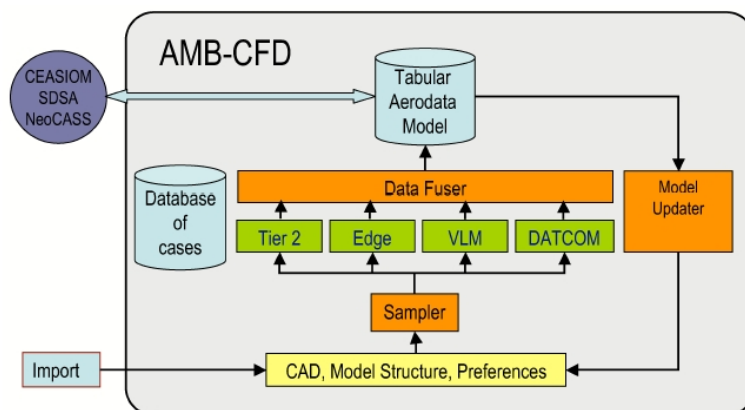


Figure 2.11: AMB CFD architecture [5]

to estimate aerodynamic derivatives based on geometric details and flight conditions. Forces and moments acting on the aircraft's individual components, such as fuselage, wing, horizontal and vertical tail, are generated by DATCOM. The input file is provided by interpreting and formatting geometric descriptions of the aircraft in AC Builder. Besides, DATCOM adds the flight conditions of interest to the data file [16].

TORNADO

TORNADO [23] is a Vortex Lattice Method (VLM) implemented as open source in Matlab. It implies a modified horse shoe vortex singularity method for computing steady and low reduced frequency at time-harmonic flows over wings [16]. The flow solver predicts a wide range of aerodynamic derivatives. Tables are generated to export the results to flight simulators and flight control system design software. TORNADO calculates first-order derivatives by small disturbances on a pre-selected state. By means of falsification in the wake, non-linear effects are visualized. Analysis in regions with interactions of the main wing and stabilizer are important. Compressibility effects at high Mach numbers are estimated by the Prandtl-Glauert similarity rule [24], whereas the zero-lift drag is obtained by Eckerts flat plate analogy [25]. The fuselage surface is formed by a number of flat panels while lifting surfaces are created as thin, square surface segments. Curved surfaces such as airfoil camber and leading edge control surfaces are modeled by surface normal rotation. Trailing edge devices are shaped by mesh deformation, so deflections are permitted during modification of horseshoe vortices. The input file of TORNADO is provided by the XML output file of AC Builder [16].

EDGE

EDGE is a CFD flow solver based on Navier-Stokes equations for viscous and inviscid compressible flow problems in unstructured grids. The flow solver is used for steady-state and time accurate calculations including maneuvers and aeroelastic simulations. Therein different turbulence models are included. EDGE creates aerodynamic tables to provide the number of possible combinations of control surfaces and corresponding deflection angles. Thereby EDGE applies the Euler mode with preset values for the maximum and minimum deflection of each control surface. The control volume is modeled by a dual grid, which is calculated by control surfaces for each edge of the input mesh. Based on boundary conditions the aerodynamic of

control surface deflections are computed. The input file of EDGE Euler is obtained by a volume mesh generated in SUMO [16].

Depending on the flight envelope of the aircraft, the applicable estimation method has to be selected by the user, see figure 2.12. In preliminary design, as long as the flight attitude remains within the limits of the flight envelope and in the range of low-speed aerodynamics, Tier 1 module predicts satisfying aerodata [5]. Furthermore, based on linear or inviscid methods this module provides good results for low angles of attack. Tier 1 module contains the methods DATCOM and TORNADO [6]. In case the flight attitudes are close to the border of the flight envelope or an improved design of the flight control system is required, Tier 1+ tool is used to predict the aerodynamic behavior. Tier 1+ module is defined by an EDGE CFD code, which is suitable to compute aerodata for transsonic flight since compressibility effects are considered. The third module Tier 2 rests upon RANS methods required for higher fidelity computations or identification of parameters in areas of nonlinear aerodynamic [5]. In figure 2.13 the disparities of the multi fidelity CFD tools are summarized. While figure 2.13a depicts the flight envelope covered by the different methods, figure 2.13b shows the fidelity of the applied methods.

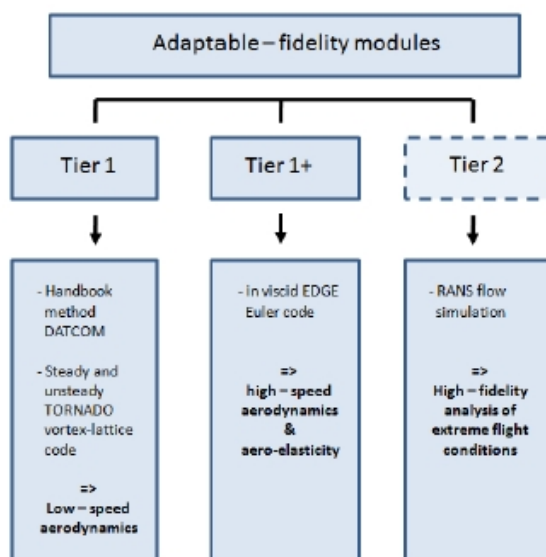


Figure 2.12: Adaptable fidelity modules [6]

2.3.4 Propulsion

Propulsion is a tool to generate the database of the engines, which is required for calculations in SDSA to serve the calculation of stability and control. The input file of Propulsion is provided by AMB and the created database is added to the XML data file [6].

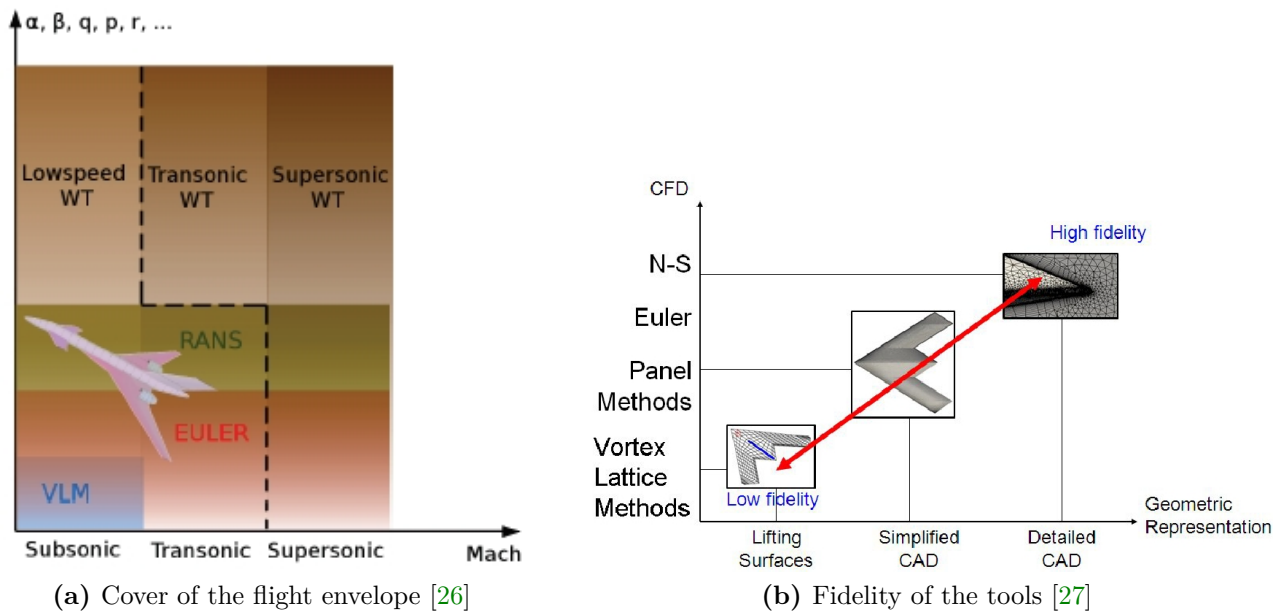


Figure 2.13: Multi fidelity CFD tools

2.3.5 SDSA

SDSA (Simulation and Dynamic Stability Analyzer) contains a code that employs a nonlinear model with equations of motion [13]. This module includes the following functionalities [28]:

1. Stability analysis
 - a) Eigenvalue analysis of linearized model
 - b) time history identification (nonlinear model)
2. Six DOF flight simulation
 - a) test flights, including trim response
 - b) turbulence
3. Flight Control System (FCS) based on Linear Quadratic Regulator (LQR) theory
4. Performance prediction
5. Miscellaneous (data review, results review, cross plots)

The structure and functionality of SDSA is visualized in figure 2.14.

In all functions of SDSA the aircraft motion is described by a mathematical nonlinear model with six DOF. Information about the implemented model are described in the SDSA documentation by SimSac [28]. For the eigenvalue analysis the model is linearized by the calculated Jacobian matrix of the state derivatives around the equilibrium point. The flight simulation module is applied for flight tests to detect flight parameters in real time. By means of detected data the modes of motion and their characteristics such as period, damping coefficient,

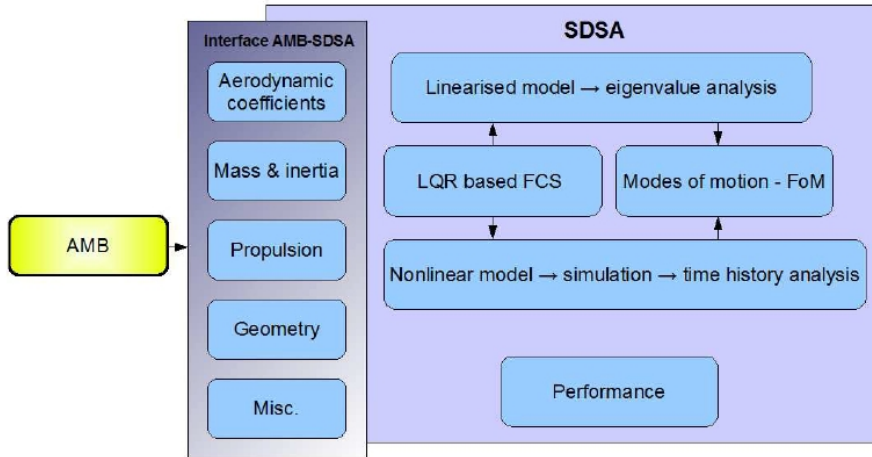


Figure 2.14: SDSA structure and functionality [5]

and phase shift are established. The results of the stability analysis are displayed as figures of performance regarding JAR/FAR, ICAO, and MIL regulations. The FCS module based on the LQR approach calculates control matrices of the total flight envelope to provide results for simulation or stability analysis. Hence, computations of stability characteristics for the closed loop case are permitted and flight simulations with FCS are accomplishable. The Performance module predicts parameters such as flight envelope, maneuvers, range, and endurance characteristics.

Starting point of the functionalities is the computation of the trimmed state. Therefore, the user has to interact with the system through a system of GUIs for initial conditions, weather conditions including blast and turbulence. Besides the initial state, disturbances and single or double step controls have to be considered for the flight test settings [28].

To analyze the stability and control characteristics of the aircraft, SDSA requires an aerodynamic database of the total flight envelope. The force and moment data are generated into the format of table 2.15 and are saved as XML file. This table created by AMB includes nine flight condition parameters as independent variables (α , Mach, β , Q, P, R, δ_e , δ_r , δ_a). The variables α , Mach, and an arbitrary variable define a set of three-dimensional look-up tables for the six aerodynamic coefficients (C_L , C_D , C_m , C_Y , C_l , C_n). Each variable should consist of two different values, so the SDSA table contains at least sixty calculated cases. To obtain more accurate results, the table implies thousands of entries [26].

In this work SDSA is used as AMB-SDSA variant integrated into CEASIOM. To conclude, SDSA generates a database of the total flight envelope to validate CFD data and to predict stability and control properties of the aircraft [26].

α	M	β	Q	P	R	δ_e	δ_r	δ_a	C_L	C_D	C_m	C_Y	C_ℓ	C_n
x	x	x	-	-	-	-	-	-	x	x	x	x	x	x
x	x	-	x	-	-	-	-	-	x	x	x	x	x	x
x	x	-	-	x	-	-	-	-	x	x	x	x	x	x
x	x	-	-	-	x	-	-	-	x	x	x	x	x	x
x	x	-	-	-	-	x	-	-	x	x	x	x	x	x
x	x	-	-	-	-	-	x	-	x	x	x	x	x	x
x	x	-	-	-	-	-	-	x	x	x	x	x	x	x

Figure 2.15: Aerodynamic database of SDSA [7]

3 CPACS to CEASIOM

The development of a wrapper between CPACS and CEASIOM bases on a previous work by Till Pfeiffer at DLR. Due to prior research a wrapper was constructed between CPACS, TORNADO and SUMO. At first, the wrapper generates *aircraft.mat*, a Matlab file containing the definition of the aircraft in a readable structure for TORNADO. In a second step *sumo.smx*, the input file of SUMO, is created. Later on the wrapper was extended to convert data from CPACS into the standard XML structure legible by CEASIOM. Therefore, *aircraft.mat* is assigned to *acproject.mat* and transformed into *aircraft.xml*. An overview about the generated files is depicted in figure 3.1. In order to use CPACS data in Matlab, the functions *xml2struct.m* and *struct2xml.m* are defined to import geometry and tool specific data in Matlab and to export the results to the CPACS file [1]. In this thesis a new script of the wrapper is developed to permit data transfer from VAMPzero to CEASIOM. The output of the wrapper should be readable not only for the AC Builder but also for SUMO and AMB as illustrated by the dashed lines in figure 2.2. Afterwards, the generated script is integrated in the wrapper of Pfeiffer.

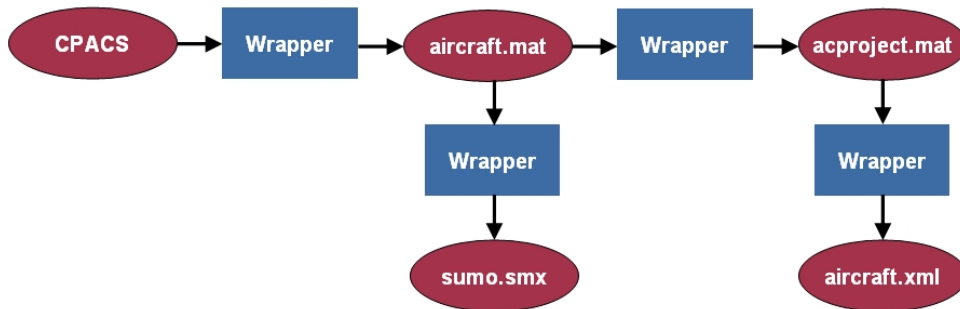


Figure 3.1: Overview of generated files

In this chapter the approach to create a wrapper between CPACS and AC Builder is presented. Moreover, difficulties to set up the wrapper are identified and the integration of airfoils and control surfaces is described. Afterwards, the wrapper is extended for data transfer from CPACS to AMB and SUMO.

3.1 CPACS to AC Builder

The objective of the wrapper is to convert the CPACS output file of VAMPzero into the input file of AC Builder as illustrated in figure 3.2. VAMPzero needs input data, comprising requirements of the aircraft to generate the output file. This input data is provided within the Matlab function *loading_acData.m* by overwriting *toolInput.xml* with data of the function *A320.m*. The function *A320.m* implies considerable information of the Airbus A320 such as payload

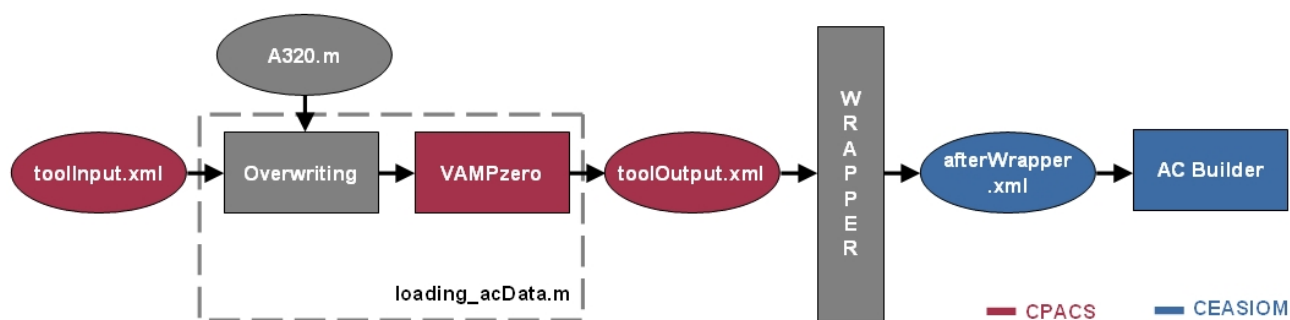


Figure 3.2: CPACS to AC Builder

and range, whereas *toolInput.xml* contains arbitrary data of an aircraft in a legible structure of VAMPzero. By using the function of a specific aircraft and integration of those data into *toolInput.xml*, the input file of VAMPzero is dynamic for several aircraft configurations. The input file of VAMPzero is stated in section 4.2. To set up a wrapper between *toolOutput.xml*, the VAMPzero output, and *afterWrapper.xml*, the input file of AC Builder, a comparison of the data provided and the data, which are still required is necessary. For that reason, the exemplary XML file of the Ranger 2000 is reviewed in order to identify the minimum number of parameters for the data transfer. The Ranger 2000 is also applied in the CEASIOM tutorial by Benoit Rey [29].

The documentation “CEASIOM XML file Definition” by Andrés Puelles [4] gives an overview of necessary input parameters to run AC Builder including associated descriptions. Following components are currently not provided by VAMPzero, but specified in the list of Puelles:

- baggage,
- cabin,
- control surfaces (aileron, flap, slat, elevator, rudder),
- engine2,
- fairings,
- fuel,
- miscellaneous,
- ventral fin,
- weight_balance,
- winglets,
- wing2.

Moreover, geometric components such as canard, tailbooms, and nacelles are not integrated in Puelles' list due to recent extensions in AC Builder. Instead of setting each parameter of the missing components to zero the presence of the components are hardcoded as zero in the wrapper. It has to be taken into account that the components baggage, cabin, miscellaneous, fuel, and `weight_balances` are always present. To avoid wrong magnitudes by the defac due to parameters which are not provided by VAMPzero, most of the parameters, listed in the documentation of Puelles, are removed from the XML file of the Ranger 2000. However, the parameters summarized in table A.1 are essential to compute the reference parameters of the wings in AC Builder. Although VAMPzero does not provide information about control surfaces these elements are inevitable for the stability and control analysis of the aircraft. The implementation of the control surfaces is described in section 3.1.3. In order to minimize the amount of parameters in the input file of AC Builder subsequent computed parameters, highlighted in green colour in the GUI, are additionally removed from the XML file. These parameters are summarized in table A.2. Attached to the thesis the file `startACBuilder.xml` contains a minimal number of required parameters for the input of AC Builder.

After specifying the required parameters the wrapper has to be generated to read the data from VAMPzero and to translate them into the input file of AC Builder. The implementation of the data transfer assumes knowledge about the definition of the components and related coordinate systems. An example for different definitions of the taper ratio in CPACS and CEASIOM is presented in section 3.1.1. Parameters that are not allocated by VAMPzero are hardcoded in the wrapper, which are displayed in section 4.3. The challenge of developing a wrapper is not only to implement the accurate translation of data but also to define a flexible environment for various aircraft configurations.

Besides geometrical descriptions of aircraft components, VAMPzero generates a weight breakdown, moments of inertia, and center of gravity. These data are not implemented in the wrapper, since AC Builder computes them automatically. In section 4.4 to 4.6 the estimated weight breakdown, moments of inertia, and center of gravity are analyzed and compared with the values of the manufacturer.

3.1.1 Taper Ratio

The taper ratio describes considerably the outer shape of the wing. Depending on the number of wing segments, depicted in figure 3.3, the taper ratio is defined at sections by the local chord divided by the root chord in equation (3.1) [4].

$$\lambda = \frac{c}{c_R} \quad (3.1)$$

This definition is applied in CEASIOM whereas in VAMPzero the local chord is related to the chord of the inner close-by section due to the definition of a local coordinate system at each section. Hence, the taper ratio describes the relative diminution of a segment, which has to be translated via equations (3.2) to (3.4) to the absolute taper ratio regarding the root chord as used in CEASIOM.

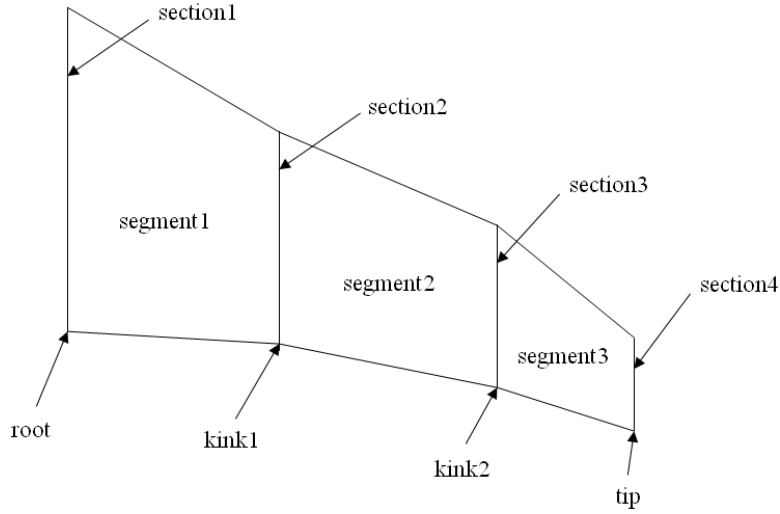


Figure 3.3: Wing planform geometry

$$\lambda_{kink1}^{abs} = \lambda_{kink1}^{loc} \quad (3.2)$$

$$\lambda_{kink2}^{abs} = \lambda_{kink1}^{loc} \cdot \lambda_{kink2}^{loc} \quad (3.3)$$

$$\lambda_{tip}^{abs} = \lambda_{kink1}^{loc} \cdot \lambda_{kink2}^{loc} \cdot \lambda_{tip}^{loc} \quad (3.4)$$

3.1.2 Airfoils

The integration of airfoils into CEASIOM is stated within the geometrical description of the wings. For example, the parameter *airfoilRoot* comprises the file name of the airfoil at the root of the wing, which is stored as a dat file in the airfoil library of CEASIOM. In order to import the point list of the generated airfoils from VAMPzero to CEASIOM, depicted in figure 3.4, the Matlab function *aeroRun.m* is developed. The function scans the airfoil library and in case the airfoils of VAMPzero are not discovered, new dat files are created and added. The airfoil is defined by an upper and lower surface, where the points within the range of 0 up to 1 are arranged in the order as presented in figure 3.5a. Otherwise, the airfoil is not able to be loaded into CEASIOM. Figure 3.5b shows exemplarily the non-symmetric NACA airfoil 653218 imported from VAMPzero into CEASIOM.

3.1.3 Control surfaces

Stability and control of the aircraft is achieved by deflecting control surfaces while the aircraft suffers disturbances or executes maneuver in flight. Aerodynamic control is obtained mainly

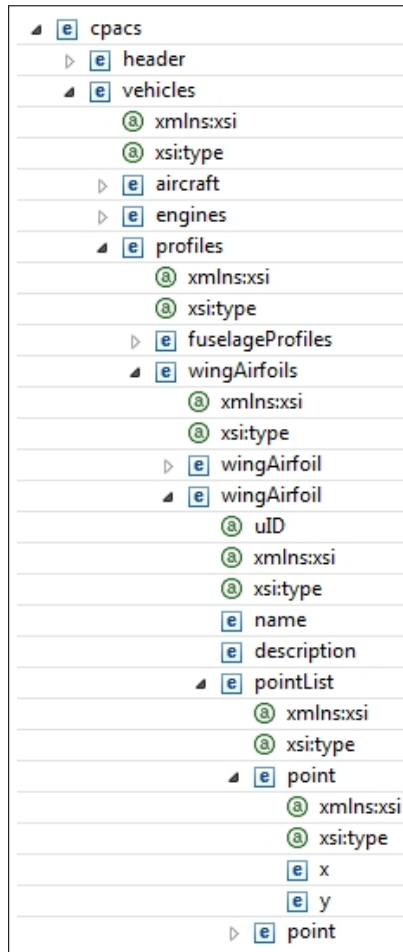
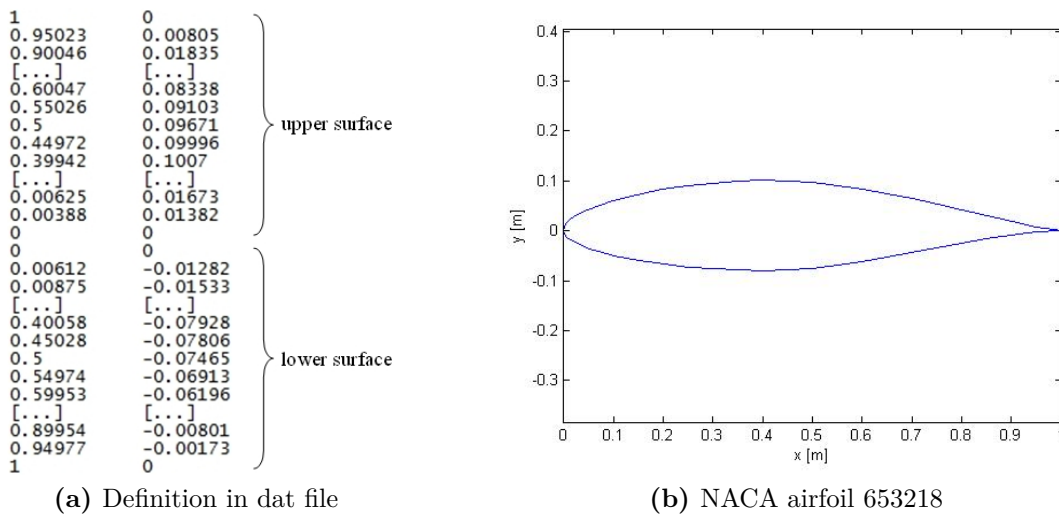


Figure 3.4: Airfoil definition in CPACS



(a) Definition in dat file

(b) NACA airfoil 653218

Figure 3.5: Imported airfoil into CEASIOM

through three sets of surfaces, which are depicted in figure 3.6. The elevator provides longitudinal control for the aircraft. The lateral control employs primarily the aileron and secondarily the rudder. Directional control is primarily achieved with the rudder and secondarily obtained with the aileron.

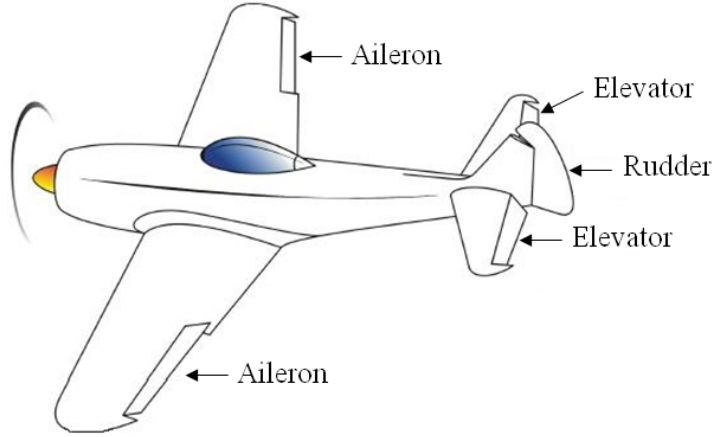


Figure 3.6: Control surfaces [8]

The control surfaces in CPACS are separately implemented at each wing, bounded by an inner and outer board. While the location of the boards along the wingspan is described by η_{LE} and η_{TE} , the chord of the control surface is specified by ξ_{LE} along the wing chord. The geometrical definition of a control surface is illustrated in figure 3.7. Figure 3.8 shows exemplarily the integration of the aileron into the CPACS structure. Although CPACS is able to include control surfaces, VAMPzero does not provide them yet. Hence, the transfer of geometrical descriptions of control surfaces from CPACS to CEASIOM is not implemented in the script. Due to previous applications of the wrapper, control surfaces are automatically added as sections to the wing. This causes errors regarding the import of aircraft data into AC Builder owing to the approvable number of sections. For that reason, control surfaces are removed from the CPACS file and are hardcoded in the wrapper instead. Table 3.1 displays the implemented data of the control surfaces based on data of the Airbus A320 used at KTH. The span of the control surface refers to the local wingspan whereas the chord obtains to the local wing chord.

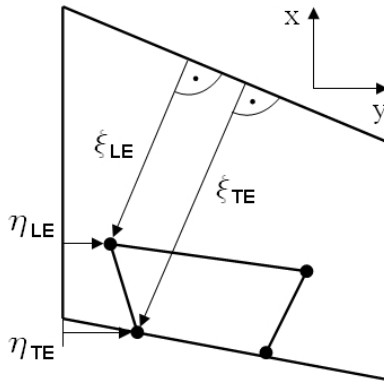


Figure 3.7: Geometric definition in CPACS [8]

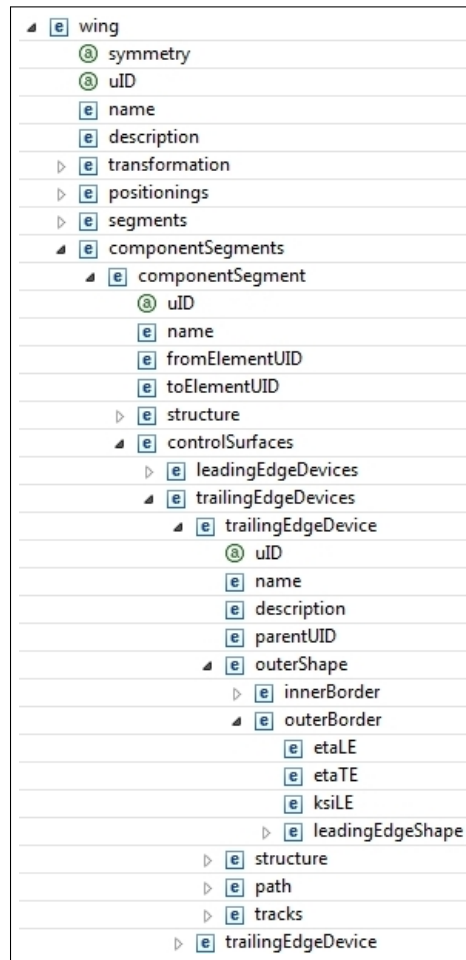


Figure 3.8: Aileron definition in CPACS

Table 3.1: Definition of control surfaces

control surface	span [%]	chord [%]	max. deflection [°]
aileron	0, 213	0, 213	+25/-10
elevator	0, 95	0, 32	+30/-15
rudder	0, 95	0, 23	±30

3.1.4 Unconventional aircraft configuration

Aerodynamic analysis of unconventional aircraft configurations within the frame of CEASIOM is of special interest to validate and improve the conceptual design tools. The DLR provides a CPACS file of a Blended Wing Body (BWB), which is used in this thesis to improve the developed wrapper. The BWB consists of nine segments and does not include a fuselage.

After conversion of BWB data from CPACS to CEASIOM the import of these data into AC Builder causes difficulties. Since it is only possible to load three semi-wing segments in AC Builder at most, some segments have to be removed from the XML file. With the function

delWingSec.m the user is able to decide which segments should be excluded. Figure 3.9 shows the input box in the Matlab Command Window for the user. In figure 3.10 the BWB is visualized with three segments incorporating spuriously a fuselage due to a permanently present fuselage in the used version of AC Builder.

```
Do you want to delete segments?  
[1] Yes (necessary for AC Builder)  
[2] No  
:1  
The current number of segments is: 9  
Include the segments number which should be deleted (e.g. [3,4,6])  
:[2,3,4,5,6,7]
```

Figure 3.9: Input box to remove segments

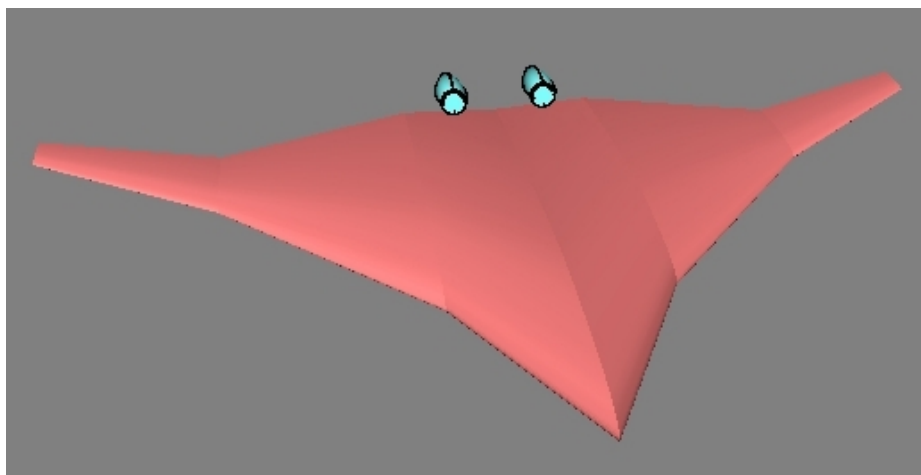


Figure 3.10: Blended Wing Body

In order to improve the accuracy of the aircraft shape, it is not feasible to integrate a rotated winglet as a further segment in the current version of AC Builder. Since the thickness of the winglet is zero it is not able to define an airfoil. Within the research project of Marchant [18] a suggestion for an extended winglet definition is made, which is included in the revised version of AC Builder allowing the definition of a fourth segment. Based on the low fidelity of the geometric shape the BWB model is only applicable for rough aerodynamic analysis. However, the weight estimation is valid for conventional aircraft configurations. The import of BWB data into SUMO is of special interest, since all wing segments are involved and display the aircraft more precisely. Moreover, the generated volume mesh in SUMO allows the application of EGDE Euler, which is suitable to predict aerodynamic forces and moments of unconventional configurations. A wrapper to translate data from CPACS to SUMO is already developed by Pfeiffer at DLR.

3.2 CPACS to AMB

The accurate description of the aircraft shape in AMB is influenced by the number of wing segments defined in the XML file. In order to achieve proper aerodynamic forces and moments for flight dynamics analysis, the wing has to be defined by three segments whereas the horizontal and the vertical tail have to consist of two segments, taking into consideration that the sections do not overlap each other. Otherwise, AMB will crash in GEO TORNADO during the generation of a mesh between infinite close sections.

In contrast to CEASIOM, VAMPzero defines the wing in two segments and the horizontal and the vertical tail comprises one segment. For that reason, the function *addWingSec.m* is established in the wrapper allowing the user to add manually sections to the wings. As depicted in figure 3.11 the Matlab Command Window displays exemplarily the current number of sections. So the user is able to decide the number and positions of further sections. The position of a section relates to the fuselage centrum at $y = 0$ and divides an existing segment into two parts, which does not change the shape of the wing.

It is possible to load the aircraft data in AMB from the output file of AC Builder or directly from the generated file of the wrapper.

```

Do you want to add sections?
The current number of segments is (htp vtp wing): 1 1 2
[1] Yes (necessary for AMB running GEO Tornado)
[2] No
:1
Insert the wing number which should be changed (e.g. [3])
:[1,2,3]
Insert new section position (e.g. [2])
:[1,1,2]

```

Figure 3.11: Input box to add sections

3.3 CPACS to SUMO

The import of aircraft data from CPACS into SUMO is feasible either by loading the output file of the wrapper or by loading CPACS data via the generated output file of AC Builder as illustrated in figure 3.12. In case the data are loaded from AC Builder the required parameters *Root_Airfoil*, *Kink_Airfoil*, and *Tip_Airfoil* describing the airfoils of the wings are not available. However, even more data are needed if the aircraft data are transferred from the wrapper to SUMO due to missing geometrical data computed in AC Builder. In Appendix A the additional required parameters in SUMO are presented. To visualize engines in SUMO the components *Nacelle1* and *Nacelle2* have to be included in *Engine1*. Figure 3.13 shows the XML structure of an engine also comprising the definition of pylons. Due to the mesh generation of a closed surface, pylons are required to connect the nacelle with the wing.

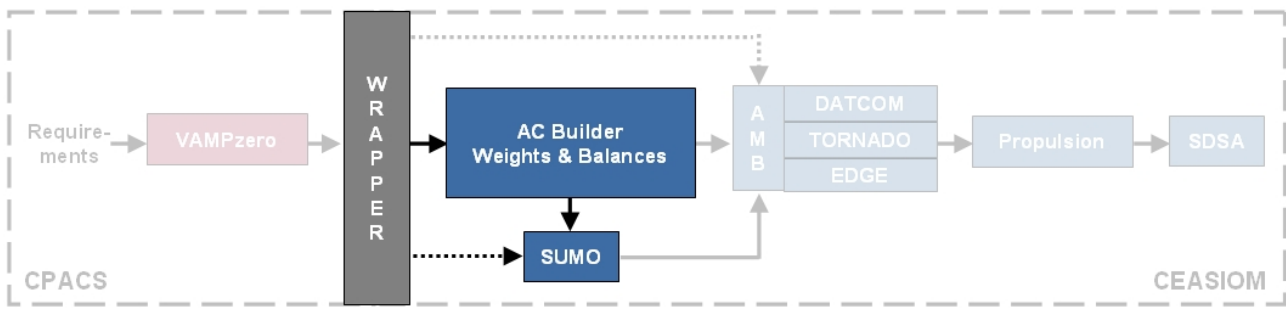


Figure 3.12: CPACS to SUMO

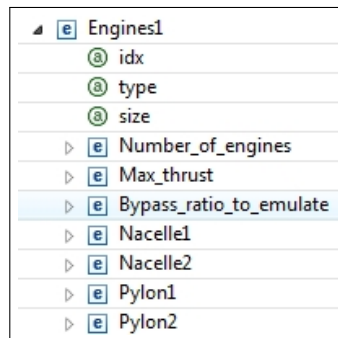


Figure 3.13: Engine definition in SUMO

One possibility to integrate parameters into the data structure of the aircraft is to add them manually to the XML file. Alternatively, AC Builder computes the required parameters except `Root_Airfoil`, `Kink_Airfoil`, and `Tip_Airfoil`. These data are adopted from a similar aircraft and are applied by the user. By involvement of the AC Builder geometric modifications of the aircraft are easily feasible. After the unique integration of the parameters into *afterWrapper.xml* future results will refer to this magnitudes, despite modifications on the aircraft in previous steps in the design process.

4 Analysis of Airbus A320

In this chapter the Airbus A320 is analyzed during the conceptual design process within CEASIOM. Firstly, the characteristics of the A320 are presented followed by a description of necessary input data of VAMPzero and their influence on estimated results. Subsequently, four shapes of the A320 are illustrated based on different data sources. The differences between the models are specified by a comparison of the magnitudes of parameters defining the aircraft components. Besides the geometric research of the A320, prediction methods of weights, moments of inertia, and center of gravity are introduced and reviewed.

As a demonstration of the conceptual design process the novel aircraft generated by the design tool VAMPzero is conducted in the frame of CEASIOM. By means of the components AC Builder, AMB, Propulsion, and SDSA the aerodynamic and flight mechanic behavior of the aircraft is determined. Moreover, the aerodynamic methods DATCOM, TORNADO, and EDGE Euler are applied to quantify their impact on design sensitivities.

The following results are generated with the CEASIOM100-v3.0_beta version supported by Matlab version R2010b under windows environment. The conceptual design tool VAMPzero is applied at revision 637.

4.1 Airbus A320

The Airbus A320 belongs to the A320 single-aisle jetliner family consisting of A318, A319, A320, and A321. The A320 is a conventional transport aircraft with two turbofans operating from international airports since 1989. With a sweep angle of 25 degrees a cruise speed of Mach 0,78 up to 0,82 is reachable. The seat capacity is 150 with a 3-3 seat arrangement in the economy class. The maximum takeoff weight (MTOW) of the A320-200 is 73500kg and has a range of 4907km, which is applied as reference configuration in this thesis. In table 4.1 the specifications of the Airbus A320 are displayed.

Table 4.1: Airbus A320 specifications [11]

Length [m]	37,57
Wingspan (without winglets) [m]	33,91
Tail height [m]	11,8
OEW [kg]	41782
MTOW [kg]	73500
Cruising speed (at 12500m) [Mach]	0,78
Maximum speed [Mach]	0,82

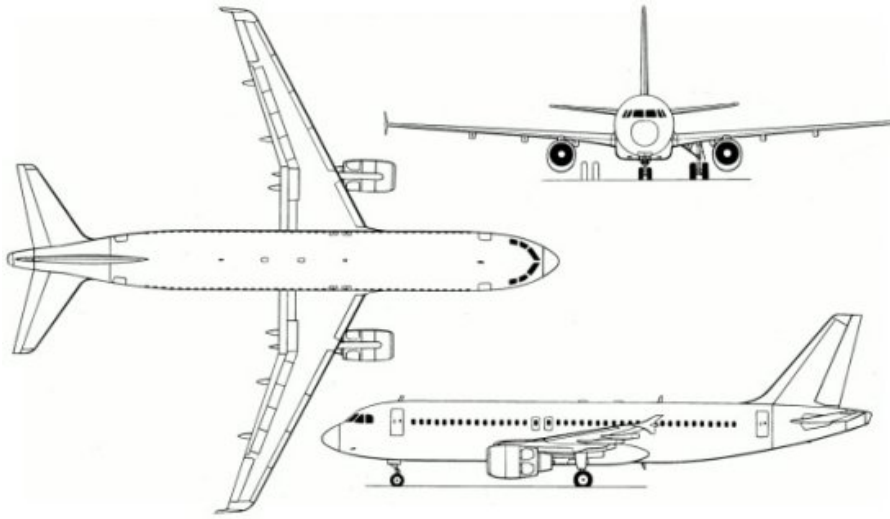


Figure 4.1: Drawing of A320 [9]

4.2 Input of VAMPzero

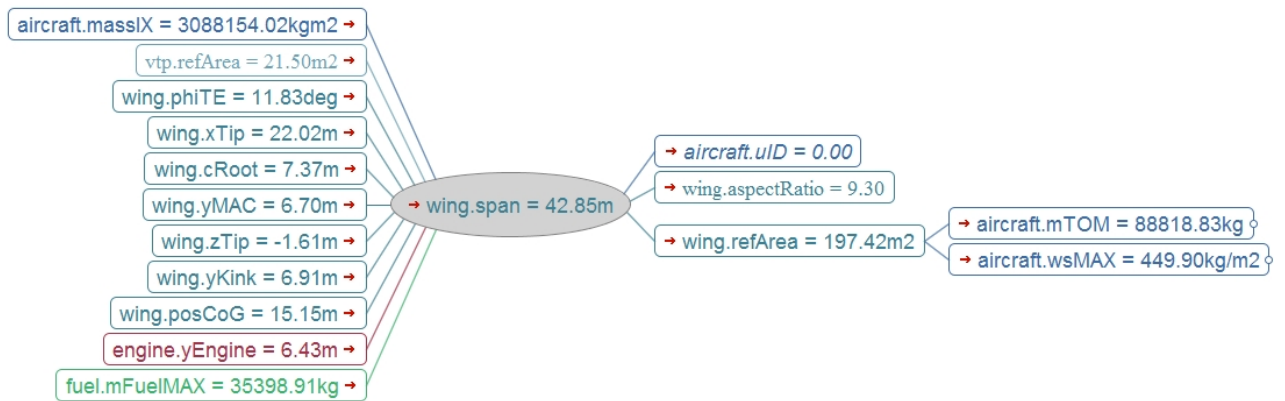
The selection of input data for VAMPzero influences the accuracy of the estimated results considerably. Beside a minimum number of input parameters obtaining convergence, additional parameters are required to improve the prediction of the aircraft design. Depending on available information the aircraft data are computed by various methods. Based on different VAMPzero input files the magnitude of the wingspan and its dependency on further parameters are exemplarily illustrated in the mindmaps of figure 4.2. In figure 4.2a the magnitude of the wingspan is displayed not taking into account the wing area in the input file. The wingspan is calculated by the aspect ratio and weights such as payload and operational empty weight (OEW). Figure 4.2b shows the magnitude of the wingspan based on the input file including the wing area = $122,4 \text{ m}^2$. The wingspan is computed by the aspect ratio and the stated wing area. Due to different considerations of the wing area the magnitude of wingspan differs up to 21%. Hence, the parameters of the VAMPzero input file have to be chosen carefully. Although an input file containing a multitude of parameters predicts the aircraft well, in most applications within the conceptual design just a few requirements are known. The input parameters of VAMPzero used in this thesis are displayed in table A.3.

4.3 Comparison of different geometries

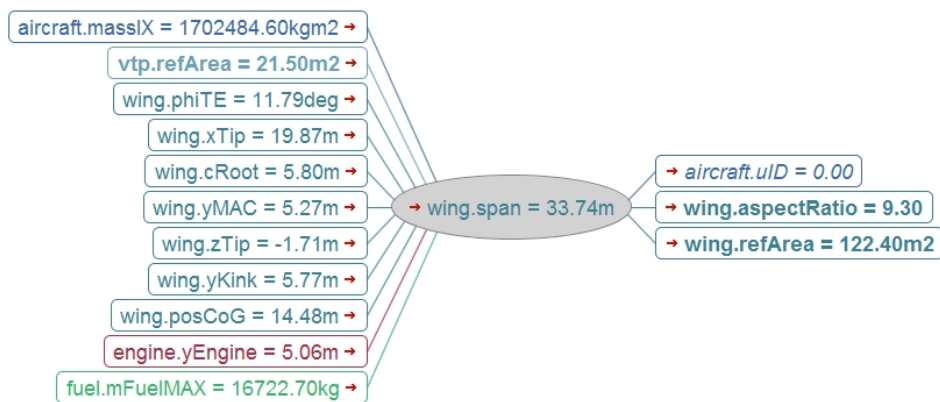
Although the Airbus A320 is currently a prevalent aircraft in the area of civil transport, several geometric information are not released by the manufacturer. In this section four shapes of the A320 are presented based on different data sources. Figure 4.3 shows the overlapping of following models:

Airbus (red line)

This model shows the A320 provided by Aerospaceweb [9].



(a) Without wing area definition



(b) Including wing area definition

Figure 4.2: Computation of wingspan**Pester (blue)**

The Pester model represents the A320 applied in the thesis “Multi-Disciplinary Conceptual Aircraft Design using CEASIOM” by Maria Pester [6]. The geometric shape is manually defined in the GUI of AC Builder taking into consideration the stated magnitudes of the manufacturer [11].

Cöllen (brown)

The Cöllen model illustrates data of the A320 after converting output data of VAMPzero into CEASIOM with the wrapper. Parameters not provided by VAMPzero, which are required in CEASIOM, are hardcoded in the wrapper. The specified magnitudes of the supplementary parameters are akin to the magnitudes in the Pester model.

VAMPzero (black line)

The VAMPzero model displays the XML output data of the conceptual design tool VAMPzero visualized in Matlab. The results are considerably affected by the input file of VAMPzero presented in table A.3 containing data of the manufacturer [11].

The overlapping of the different models shows the discrepancy between the designs. The shape of the Cöllen model coincides well with the design of the VAMPzero model, due to a similar

data set before and after translation of CPACS data into CEASIOM by the wrapper. In figure 4.3a it is conspicuous that the vertical tailplane of the VAMPzero model is located higher than in the Cöllen model. This is caused by a required modification of the vertical tailplane in AC Builder as described in section 4.7. Figure 4.3b depicts the models in front view, where different dihedrals of the main wing are illustrated. In comparison to the Pester model and Airbus model, the models based on VAMPzero data hold an dihedral angle of zero degree. Moreover, the geometry of the main wing of the VAMPzero model and Cöllen model coincides with the shape of the Pester model as depicted in figure 4.3c. However, the design of the main wing of the Pester model differs from the shape of the Airbus model due to the availability of different data sets.

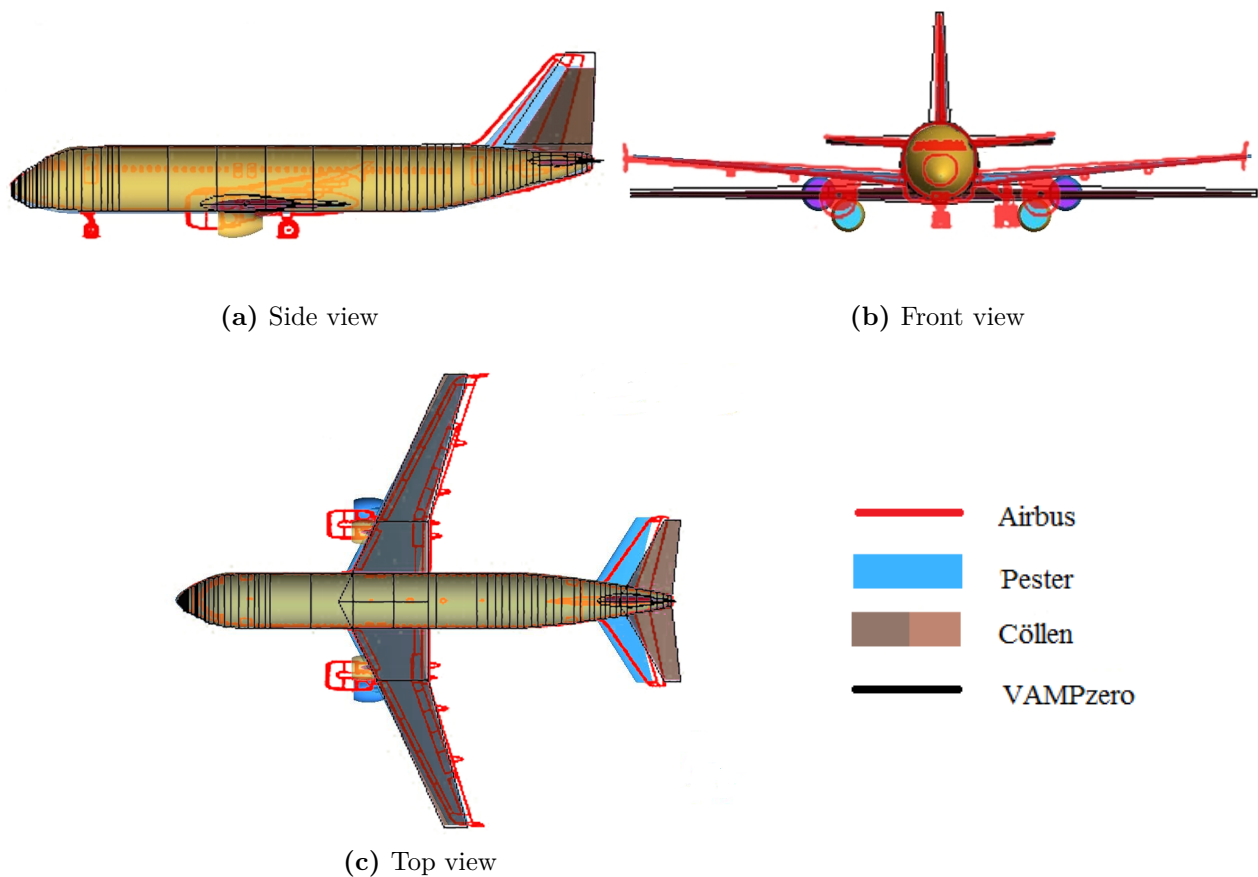


Figure 4.3: Comparison of geometries

In order to specify the geometric differences of the models, input data of the aircraft components are presented in the following sections. Since data of the Pester model and data of the Cöllen model are available in the same format, these information are considered. Moreover, some magnitudes of the Airbus model are available. Displayed magnitudes are approximated up to two numbers after the decimal point and the deviation refers to the magnitudes of the Cöllen model regarding the Pester model. Bracketed values denote hardcoded magnitudes of the wrapper.

4.3.1 Fuselage

Table 4.2 shows the input parameters of the fuselage in AC Builder within the frame of CEA-SIOM and includes available data of the Airbus model. In CPACS the fuselage is defined by a multitude of positioned sections with different cross-sectional areas. Since the nose and the tail of the aircraft are not implemented in the wrapper, these data required for the input of AC Builder are hardcoded in the wrapper. The length of the fuselage is determined by summation of the distances between sections. As shown by the magnitudes of the Cöllen model, the length of the fuselage is sufficiently approximated by VAMPzero and translated to AC Builder using the wrapper.

Table 4.2: Input parameters of the fuselage

Parameter	Airbus	Pester	Cöllen	Deviation [%]
omega_nose [°]	–	35,00	(35,00)	–
phi_nose [°]	–	12,00	(12,00)	–
epsilon_nose	–	0,82	(0,82)	–
Forefuse_vertical_diameter [m]	3,94	4,14	4,15	+0,17
Forefuse_horizontal_diameter [m]	3,94	3,95	4,15	+5,01
Forefuse_distortion_coefficient	–	0,50	(0,50)	–
omega_tail [°]	–	13,00	(13,00)	–
phi_tail [°]	–	6,00	(6,00)	–
epsilon_tail	–	2,56	(2,56)	–
Aftfuse_vertical_diameter [m]	3,94	4,14	4,15	+0,17
Aftfuse_horizontal_diameter [m]	3,94	3,95	4,15	+5,01
Aftfuse_distortion_coefficient	–	0,50	(0,50)	–
fraction_fore	–	0,50	(0,50)	–
shift_fore [m]	–	0,00	(0,00)	–
Total_fuselage_length [m]	37,57	37,57	36,82	–2,00

The deflection between the magnitude of the vertical and horizontal diameter of the fuselage in Pester and Cöllen is caused by different geometries of the cross sections. While in the Pester model the cross section of the fuselage is defined as an ellipse, the translated cross section from VAMPzero to AC Builder is described as a circle. In CPACS the cross section is defined by a point list, frequently shaping a circle but also modeling an arbitrary geometry with splines scalable in x and z-direction. The generated CPACS file in VAMPzero comprises an elliptic cross section of the fuselage. In order to handle various geometries with the wrapper, the biggest diameter is selected to define the size of the circle. For that reason, the displayed magnitudes of the vertical and horizontal diameter of the Cöllen model coincide although VAMPzero computed an ellipse.

4.3.2 Wing

Table 4.3 shows the magnitudes of the input parameters of the wing, valid for a supplemental section at two meters. Further parameters related to the wing are displayed in Appendix B. The magnitudes of the wing area and the wingspan of the Cöllen model coincide with the values of the Pester model, caused by the parameters listed in the input file of VAMPzero. Otherwise, the values differ strongly from the magnitudes of Pester, as described in section 4.2. The parameters *incidence* and *dihedral* are computed in VAMPzero to zero degree and are translated into AC Builder.

Table 4.3: Input parameters of the wing
* defined in VAMPzero input file

Parameter	Airbus	Pester	Cöllen	Deviation [%]
area [m ²]	122,4	122,40	122,40*	+0,00
Span [m]	34,10	33,91	33,74*	-0,51
spanwise_kink1	—	0,11	0,12	+7,63
spanwise_kink2	—	0,37	0,35	-6,29
taper_kink1	—	0,88	0,86	-2,410
taper_kink2	—	0,62	0,58	-5,72
taper_tip	—	0,25	0,25*	-1,54
root_incidence [°]	—	-1,25	0,00	—
kink1_incidence [°]	—	-1,25	0,00	—
kink2_incidence [°]	—	0,51	0,00	—
tip_incidence [°]	—	-1,38	0,00	—
LE_sweep_inboard [°]	25,00	27,00	25,06*	-8,62
LE_sweep_midboard [°]	25,00	27,00	25,06*	-8,62
LE_sweep_outboard [°]	25,00	27,00	25,06*	-8,62
dihedral_inboard [°]	—	5,00	0,00	—
dihedral_midboard [°]	—	5,00	0,00	—
dihedral_outboard [°]	—	5,00	0,00	—
placement	—	0,22	0,09	-60,66
apex_locale	—	0,31	0,33	+4,00

The deviation of *spanwise_kink* is caused by the manual input of the user to add further sections to the wing. The parameter *spanwise_kink* is defined in equation (4.1), where the spanwise kink location y_{kink} measures the distance of the kink to the plane of symmetry at $y = 0$ [4]. If the user defines the spanwise location at kink1 as approximately 1,86m the deviation approaches to zero. The spanwise kink2 location at 5,90m is computed by VAMPzero considering two wing segments. Depending on whether the position of the inserted section is greater or less than 5,90m, the deviation of *spanwise_kink1* or *spanwise_kink2* changes. In contrast to the

spanwise location the wingspan influences marginally the deviation of *spanwise_kink*, since the wingspan of the Cöllén model agrees well with the magnitude of the Pester model.

$$spanwise_kink = \frac{2y_{kink}}{b} \quad (4.1)$$

Moreover, the position of the additional section influences the deviation of *taper_kink*. The parameter *taper_kink* is defined in equation (4.2), where the local chord c and the root chord c_R correspond to the airfoil scaling in x-direction within CPACS. Despite the insertion of the additional segment, the wrapper considers a constant gradient on the leading edge of the wing. So the additional *taper_kink* does not influence the wing shape. Depending on whether the position of the inserted section is greater or less than 5,90m, the deviation of *taper_kink1* or *taper_kink2* changes. The parameter *taper_tip* is independent of the location of the additional section, so magnitude of the Cöllén model matches the value of Pester. The tip chord is generated in VAMPzero with respect to the wingspan. In summary, the deviation of *taper_kink* is mainly caused by the airfoil scaling computed by VAMPzero. Furthermore, the input of the additional section influences the values, but does not change the shape of the wing.

$$taper_kink = \frac{c}{c_R} \quad (4.2)$$

Important parameters for the positioning of the wing are *placement* and *apex_locale*. The parameter *placement* in equation (4.3) describes the vertical location of the wing. In equation (4.4) the parameter d_p constitutes the distance between the lowest cross section point of the fuselage and the root chord of the leading edge [4]. The translation of the wing in z-direction z_{wing} is calculated in VAMPzero regarding the center of the fuselage at $z = 0$. The strong deviation of *placement* between the magnitude of the Cöllén model and Pester model is mainly influenced by the z-translation, generated by VAMPzero. The *Aftfuse_vertical_diameter* affects marginally the deviation of *placement*, due to the deviation of 0,17% as displayed in table 4.2.

$$placement = \frac{d_p}{Aftfuse_vertical_diameter} \quad (4.3)$$

$$d_p = 0,5 \cdot Aftfuse_vertical_diameter - |z_{wing}| \quad (4.4)$$

The parameter *apex_locale* in equation (4.5) describes the horizontal location of the wing. The deviation of *apex_locale* is not only influenced by the deviation of *Total_fuselage_length*, as displayed in table 4.2, but also by the x-translation of the wing, defined in the input file of VAMPzero.

$$apex_locale = \frac{x_{wing}}{Total_fuselage_length} \quad (4.5)$$

4.3.3 Horizontal tail

In table 4.4 the magnitudes of the input parameters of the horizontal tail are shown, valid for a supplemental section at one meter. Further parameters related to the horizontal tail are displayed in Appendix B. The analysis of the deviation between the magnitudes of the Cöllén model and Pester model is referred to the analysis of the wing in section 4.3.2.

Table 4.4: Parameters of the horizontal tail

* defined in VAMPzero input file

Parameter	Airbus	Pester	Cöllén	Deviation [%]
area [m ²]	–	31,00	31,00*	+0,00
Span [m]	12,44	12,45	12,01*	–3,56
taper_tip	–	0,33	0,30*	–8,18
root_incidence [°]	–	0,00	0,00	0,00
kink_incidence [°]	–	0,00	0,00	0,00
tip_incidence [°]	–	0,00	0,00	0,00
LE_sweep_inboard [°]	–	32,32	30,13*	–6,77
LE_sweep_outboard [°]	–	32,32	30,13*	–6,77
dihedral_inboard [°]	–	5,00	0,00	–
dihedral_outboard [°]	–	5,00	0,00	–
apex_locale	–	0,82	0,89	+8,82

4.3.4 Vertical tail

Table 4.5 shows the magnitudes of the input parameters of the vertical tail, valid for a supplemental section at one meter. Further parameters related to the vertical tail are displayed in Appendix B. The analysis of the deviation between the magnitudes of the Cöllén model and Pester model is referred to the analysis of the wing in section 4.3.2.

4.3.5 Engine

In table 4.6 the input parameters of the engine are shown. Further parameters related to the engine are displayed in Appendix B. The deviation of Y_locale is in comparison to X_locale relatively large. The parameter Y_locale is defined in equation (4.6). The discrepancy of Y_locale is mainly caused by the translation of the engine in y-direction calculated in VAMP-zero. The wingspan induces a minor deviation with approximately 1,96% due to the deviation of the wingspan, as displayed in table 4.3.

$$Y_locale = \frac{2y_{engine}}{b} \quad (4.6)$$

Table 4.5: Parameters of the vertical tail
* defined in VAMPzero input file

Parameter	Airbus	Pester	Cöllen	Deviation [%]
area [m ²]	–	21,50	21,50*	0,00
Span [m]	5,80	5,87	5,80*	–1,11
taper_tip	–	0,35	0,33*	–4,30
root_incidence [°]	–	0,00	0,00	0,00
kink_incidence [°]	–	0,00	0,00	0,00
tip_incidence [°]	–	0,00	0,00	0,00
LE_sweep_inboard [°]	–	40,40	34,43*	–14,77
LE_sweep_outboard [°]	–	40,40	34,43*	–14,77
dihedral_inboard [°]	–	0,00	0,00	0,00
dihedral_outboard [°]	–	0,00	0,00	0,00
apex_locale	–	0,80	0,85	+6,53

The magnitude of *Propeller_diameter* is negligible, since the engine is implemented as a turbofan without propeller. The magnitude of the computed thrust in VAMPzero differs from the magnitude in the Pester model. The displayed thrust of the Pester model is akin to the thrust at 112 kN of the engine designation CFM56-5-A1 as stated in the Engine Yearbook 2007 [30]. As depicted in the mindmap of figure 4.4, the estimated thrust in VAMPzero refers to one engine although the dependency on two engines is displayed.

Further required input parameters for CEASIOM such as *toe_in* and *Thrust_reverser_effectivness* are not provided by VAMPzero, but hardcoded in the wrapper.

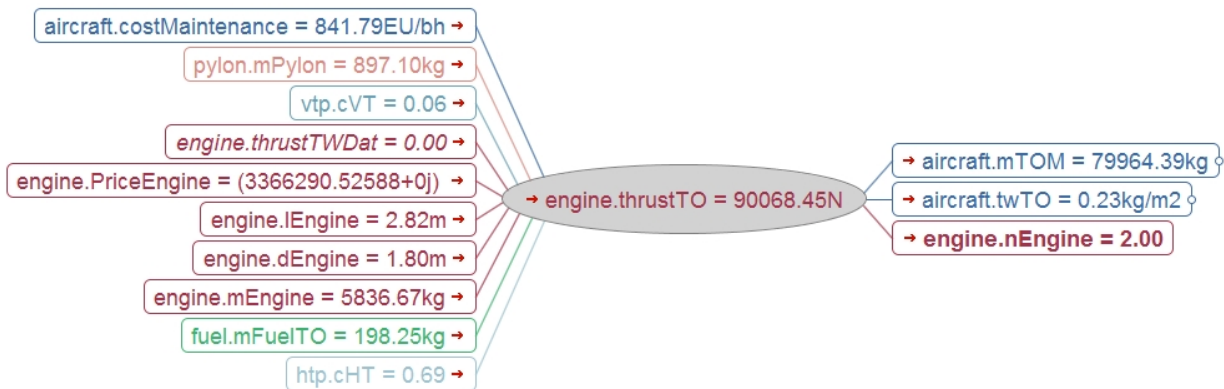


Figure 4.4: Thrust at takeoff for one engine

Table 4.6: Parameters of the engine
* defined in VAMPzero input file

Parameter	Pester	Cöllen	Deviation [%]
X_locale [m]	0,35	0,35	-0,43
Y_locale [m]	0,40	0,30	-24,81
Z_locale [m]	0,00	(0,00)	-
fineness_ratio	1,66	1,56	-6,00
d_max [m]	1,90	1,80	-5,26
toe_in [°]	0,00	(0,00)	-
pitch [°]	2,00	(0,00)	-
Thrust_to_weight_ratio	0,40	(0,00)	-
Propeller_diameter [m]	1,276	(1,98)	-
Max_thrust [kN]	104,00	90,07	-13,39
Bypass_ratio	6,00	5,90*	-1,67
Thrust_reverser_effectivness	0,30	(0,30)	-
Fan_cowl_length_ratio	0,00	(0,00)	-

4.4 Weight breakdown

Accurate weight estimation is necessary to predict the performance and flight handling properties of the aircraft. The computation of the weight is a critical part in the conceptual design process reviewed in this section.

AC Builder contains the function *Weights and Balances* wherein weights, moments of inertia, and centers of gravity of the aircraft are computed. The weight estimation bases on a universal method for various aircraft configurations, as documented in the doctoral thesis of Isikveren [17]. Furthermore, the current version of CEASIOM includes the stand-alone tool *Weights and Balances* allowing the user to decide among five weight estimation methods:

- Cessna
- Howe
- Raymer
- USAF
- Torenbeek

These methods are analyzed in the thesis “Further Development and Testing of CEASIOM for Weight, Balance and Structural Design of Aircraft” by Javier Muñoz Martín in which advantages and disadvantages of the different weight predictions are presented [31]. In comparison to the integrated function *Weights and Balances* of AC Builder the stand-alone tool *Weights and Balances* allows the selection of an applicable method for the aircraft configuration by the

user. This design tool is not mandatory in the conceptual design process of CEASIOM, but suitable to improve the accuracy of the weight prediction.

Figures 4.5 and 4.6 show the results of the stand-alone tool *Weights and Balances*. The pie charts depict the weight breakdown of the five weight estimation methods with regard to the Cöllen model. Besides geometric data of the aircraft, the Howe method needs further parameters and constant values. The very large weight proportion of the fuselage depicted in figure 4.6 is caused by the required constant values, which are displayed in Appendix C. Moreover, information about input parameters and a detailed weight breakdown of each estimation method is presented.

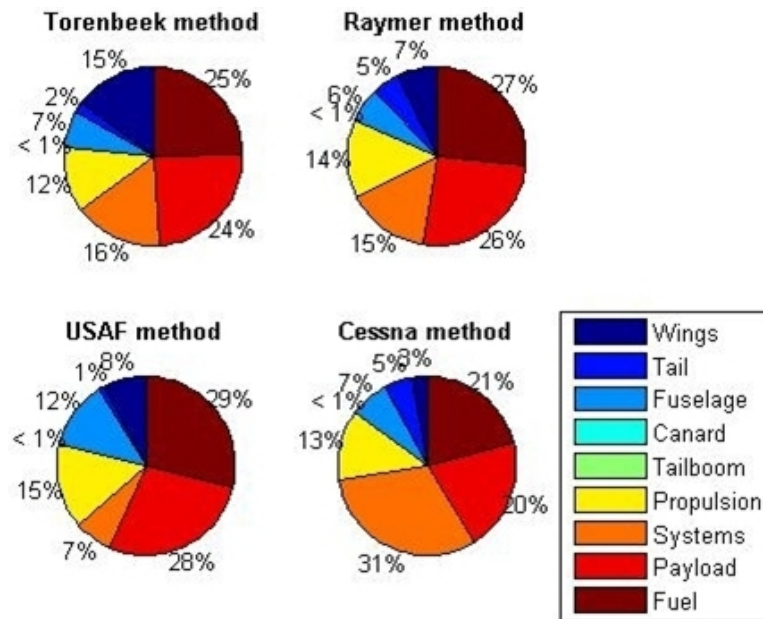


Figure 4.5: Results of the stand-alone tool *Weights and Balances*

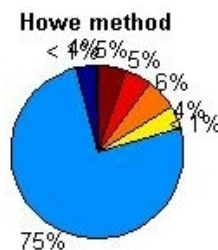


Figure 4.6: Result of the Howe method

The selection of the applicable weight estimation method is supported by the knowledge of the weight breakdown of a medium-range aircraft depicted in figure 4.7. These data base on a LTH Mass Analysis of Airbus [32] and on documents of the lecture “Aircraft Design” of Prof. Gollnick at Hamburg University of Technology [33]. The presented weight breakdown is estimated by the average weight of the components of a short and long range aircraft. In order to compare the weight breakdown of the medium-range aircraft with the results in figure 4.5

easily, the component *Others* is introduced comprising additional weights such as equipment and landing gear. The system weight is not considered in the weight breakdown. A comparison between the pie charts of figure 4.5 and 4.7 shows that the weight breakdown of the Torenbeek method is similar to the weight breakdown of the medium-range aircraft. Especially the weight proportion of wings, fuel, and payload coincide well. Hence, the Torenbeek method is selected to estimate the weight of the aircraft, which is employed in the further conceptual design process.

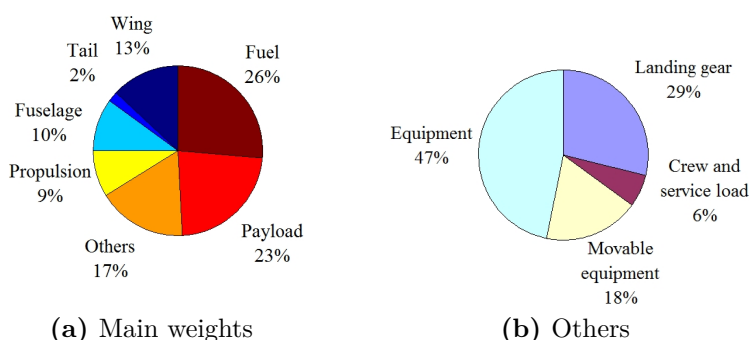


Figure 4.7: Weight breakdown of a medium-range aircraft

Table 4.9 implies the weight breakdown of the Airbus A320 including magnitudes of the manufacturer and results of VAMPzero, AC Builder, and Torenbeek. The model and method nomenclature displayed are specified in table 4.7 and 4.8, which are supportively illustrated in figure 4.8. The input data of AC Builder and *Weights and Balances* rests upon the output file of VAMPzero. The weight breakdown is structured in the order of the weight definitions in the doctoral thesis of Isikveren [17]. The displayed magnitudes of the weights are rounded off. In order to evaluate the accuracy of the estimated weights, the weight components are reviewed separately in following sections. Concluding, the results are compared with the magnitudes provided by the manufacturer [11] and the weights of the Pester model.

Table 4.7: Nomenclature of different models

model	description
Airbus	data provided by Airbus and Aerospaceweb [9]
Pester	data applied in the thesis by Pester [6]
VAMPzero	data of VAMPzero before the wrapper, <i>toolOutput.xml</i>
Cöllen	data after the wrapper

Table 4.8: Nomenclature of different methods

method	description
VAMPzero	computation within VAMPzero
AC Builder	computation within AC Builder, <i>afterACBuilder.xml</i>
Torenbeek	computation based on the Torenbeek method, <i>afterWB.xml</i>

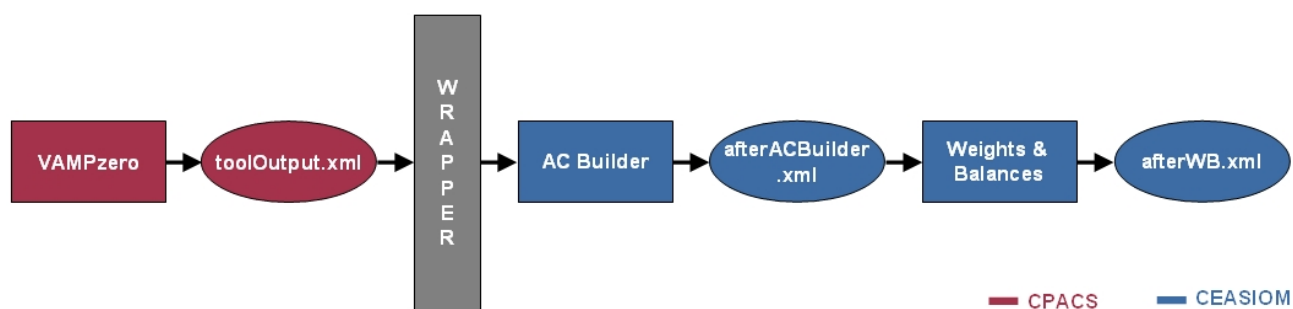


Figure 4.8: Extract of the analysis chain

4.4.1 Structure Weight

The structure weight of the aircraft consists of the weight summation of the components wing, winglets, horizontal tail, vertical tail, dorsal and ventral fin, fuselage, and landing gear as described in equation (4.7). The Pester model includes, in contrast to the other models winglets, but the winglet weight and load penalty are zero. In all models the dorsal fin and ventral fin are not present.

$$W_{stru} = W_{wing} + \Delta W_{wlet} + W_{htp} + W_{vtp} + W_{fins} + W_{fus} + W_{lg} \quad (4.7)$$

The discrepancy of the structure weight in table 4.9 between the Pester model and Cöllen model is caused by different geometric input parameters and additional factors. For example, the reduced span of wing, horizontal tail and vertical tail, as displayed in table 4.3 to 4.5, influences besides other parameters the decrease of the component weights in the Cöllen model. Nevertheless, it is unclear why the weight of the main wing differs that strongly. The estimated structure weight of the VAMPzero model coincides well with the result of the Pester model.

4.4.2 System Weight

The system weight of the aircraft consists of the summation of the system weights fuel, flight control, APU, instrumentation, avionic, hydraulic, electric, ECS, green furnishing, and miscellaneous as described in

$$W_{sys} = W_{sfuel} + W_{fcnt} + W_{apu} + W_{inst} + W_{avn} + W_{hyd} + W_{elec} + W_{ecs} + W_{furn} + W_{misc}. \quad (4.8)$$

In contrast to VAMPzero and the tool *Weights and Balances*, the XML files of the AC Builder method do not imply the system weight. The denoted system weight in the XML file of VAMPzero does not consider the weight of furnishing, which is added to the displayed magnitude in table 4.9 manually. Detailed weight predictions for each system component are able to be estimated by using methods of Raymer [12] and Torenbeek [34]. In order to determine a rough conceptual weight prediction of the total systems group, the constant weight passenger coef-

Table 4.9: Weight breakdown of A320 [kg]
* defined in VAMPzero/AC Builder input file

Model Method	Airbus -	Pester AC Builder	VAMPzero VAMPzero	Cöllen AC Builder	Cöllen Torenbeek
Wing	–	8767	9240	7498	9118
Horizontal tail	–	844	760	723	665
Vertical tail	–	490	540	403	418
Fuselage	–	7208	8724	7327	4213
Main landing gear	–	3116	3057	2466	2521
Aux landing gear	–	2547	–	–	–
Total structure	-	22972	22321	18417	16935
Total systems	-	-	7771	-	9965
Pylons	–	449	897	317	–
Engines	4531	6647	5837	4155	–
Nacelles	–	2139	–	1337	–
Total propulsion system	-	9235	6734	5809	7239
Compl allowance plus paint	–	5727	–	5727	–
Operating items	–	939	5484	872	–
OEW (calculated)	–	38873	42310	30825	34139
OEW (displayed)	41244	48123	42310	40076	34139
Deviation OEW		+17%	+3%	-3%	-17%
Payload	19256	15311	19220*	14968	14850
MZFW	60500	63434	61530	55044	48989
Fuel	18729	12500	18434	15311*	15311
MTOW (calculated)	79229	75934	79964	70355	64300
MTOW (displayed)	73500	73397	79964	58724	64300
Deviation MTOW		-0%	+9%	-20%	-13%

efficient of Scott and Nguyen [35] is identified based on statistical correlations [17]. A simple estimation for the total system weight is

$$W_{sys} = 0,6\Pi_{ATM}k_{pax}PAX \quad (4.9)$$

where PAX describes the design number of passengers accommodated in the aircraft. The parameter

$$\Pi_{ATM} = e^{2,965-0,001525YEIS} \quad (4.10)$$

depends on the year of entry into service (YEIS) of the Airbus A320 in 1988. Moreover, the constant weight passenger coefficient is defined as

$$k_{pax} = k_{cpax} + \Phi_{acsel} (k_{bpax} - k_{cpax}) \quad (4.11)$$

including

$$k_{bpax} = 15,651 + 38,27 \frac{V_{cab}}{PAX} + 4297,1 \frac{d_{ecab}}{PAX} \left(1 - \frac{4,7923}{l_{cab}}\right) \quad (4.12)$$

and

$$k_{cpax} = 55,168 + 10,344N_{abs} + 13,952\Phi_{pax} + 2,616\Phi_{pax}N_{abs} - 3,98PAX^{0,3494}. \quad (4.13)$$

The equivalent cabin diameter is formulated by

$$d_{ecab} = \frac{2h_{cab} + w_{cab}}{3}. \quad (4.14)$$

With available XML data of the Cöllen model the total system weight is approximated to 9891kg, which corresponds to the magnitude of the estimated system weight of Torenbeek method. Within the computations the parameter Φ_{acsel} is considered to one.

4.4.3 Powerplant Weight

The powerplant weight consists of the weight summation of the components engines, propeller, nacelles, and pylons described in equation (4.15). Since the engines are constructed as turbofans, the propeller weight is zero. Owing to different geometric input parameters and further aspects, the powerplant weight of the Pester model and Cöllen model differs as shown in table 4.9. In table 4.6 the magnitudes of engine diameter and engine length are less in the Cöllen model than the magnitudes in Pester, leading to a reduced engine weight in Cöllen. Furthermore, the estimated pylon weight is comparatively small, due to a shortened distance between pylon and wing caused by dihedral angles at zero degrees. In VAMPzero the nacelles weight is included in the engines weight.

$$W_{pow} = W_{eng} + W_{pyl} + W_{prop} + W_{nac} \quad (4.15)$$

4.4.4 Operating Items Weight

The operating items weight of the aircraft consists of the weight contribution of flight and cabin crew including appropriate baggage, galley inserts, galley supplies, consumables, portable

water, toilet chemicals and lavatory supplies, unusable fuel, engine oil, flight manuals, over-water provisions, tow bar, ladder, and any other miscellaneous equipment. The summation of the component weights is described in equation (4.16).

$$W_{oper} = W_{fcrew} + W_{fatt} + W_{cons} + W_{fusu} + W_{misc} \quad (4.16)$$

4.4.5 Operational Empty Weight

The operational empty weight (OEW) consists of the weight summation of the structure, systems, powerplants, components, and operational items weights described in equation (4.17). In table 4.9 a calculated and displayed OEW is shown. The calculated OEW considers the magnitudes depicted in the table whereas the displayed OEW implies the magnitude of the XML file computed by the design tools. The calculated OEW does not coincide with the displayed OEW neither in Pester nor in Cöllen (AC Builder), which might be caused by the non-displayed system weight. Even though the system weight is not included in the XML file it is apparently integrated in the displayed OEW. The estimated system weight in section 4.4.2 is in accordance with the discrepancy between the calculated and displayed OEW in the Cöllen model.

$$OEW = W_{stru} + W_{sys} + W_{pow} + W_{comp} + W_{oper} \quad (4.17)$$

4.4.6 Maximum Zero-Fuel Weight

The maximum zero-fuel weight (MZFW) consists of the OEW and the maximum payload as described in

$$MZFW = OEW + W_{pay}. \quad (4.18)$$

4.4.7 Maximum Takeoff Weight

The maximum takeoff weight (MTOW) comprises three functional weight groups consisting of MZFW, fuel weight, and fuel decrement as described in equation (4.19). The design parameter fuel decrement is not considered and is set to zero. Depending on the design range the estimation of the fuel weight is a critical part of an accurate weight prediction. As an example the fuel weight of the manufacturer stated in table 4.9 includes additionally a fuel reserve weight. In table 4.9 a calculated and displayed MTOW is shown. The deviation of the calculated and displayed MTOW of the Pester model and Cöllen model (AC Builder) depends on the integration of the fuel weight. Although a fuel weight of 15311kg is stated in the Cöllen model (AC Builder) only 3726kg are taken into account in the calculation of MTOW. Hence, the MTOW of the Cöllen model (AC Builder) differs 20% to the magnitude of the manufacturer in spite of a good estimation of the OEW with a deviation of 3%. The strong deviation of the MTOW is caused by a bug in AC Builder, which will be corrected in a revised version.

$$MTOW = MZFW + W_{fuel} - \Delta W_{decr} \quad (4.19)$$

4.4.8 Conclusion

A comparison of the weight prediction tool AC Builder and the stand-alone tool *Weights and Balances* using the Torenbeek method shows that the Torenbeek method improves the weight estimation of AC Builder. The deviation between the computed MTOW of the Cöllen model and the inherent magnitude of the manufacturer is reduced from -20% up to -13%. However, large weight fluctuations occur within the weight prediction process as in the OEW estimation, where the Torenbeek method even distorts the result. Especially the structure weight in AC Builder and *Weights and Balances* is not as well predicted as in VAMPzero. With 9% deviation from the magnitude of the manufacturer the MTOW is sufficiently predicted in VAMPzero. The MTOW of the Pester model coincides with the magnitude of the manufacturer due to a similar A320 geometric input file, in contrast to a selected quantity of input parameters for VAMPzero.

4.5 Moments of inertia

Besides the total aircraft weight the distribution of the weight influences the flight dynamic behavior, reflecting moments of inertia. The moments of inertia are predictable by different methods such as Radius of Gyration, Lumped Masses, CAD Solid Modeling, Empirical Inertia Equation, and DATCOM method reviewed in the thesis of Martín [31]. The stand-alone component *Weights and Balances* estimates moments of inertia with DATCOM method. Table 4.10 implies the moments of inertia of the different prediction modules. The displayed deviations refer to the magnitudes of the Pester model, since reference data of the manufacturer are not available. The computed magnitudes of the Cöllen model (AC Builder) and Cöllen model (Torenbeek) show that Torenbeek method partly adulterates the results. VAMPzero predicts the inertia I_{xx} with 1% deviation to the magnitude of the Pester model well. However, the inertia I_{yy} and I_{zz} differs with 53% and 37% strongly from the magnitudes of the Pester model.

Furthermore, VAMPzero generates moments of inertia of the engines and Torenbeek method provides supplementary the inertia I_{xz} , significant for non symmetrical aircraft configurations.

Table 4.10: Moments of inertia [kgm^2]

Model Method	Pester AC Builder	VAMPzero VAMPzero	Cöllen AC Builder	Cöllen Torenbeek
I_{xx}	1693312	1702478	1546896	2237778
Deviation		+1%	-9%	+32%
I_{yy}	2441460	3728004	2164089	2167253
Deviation		+53%	-11%	-11%
I_{zz}	3924823	5362383	3526757	4257863
Deviation		+37%	-10%	+8%

4.6 Center of gravity

In the conceptual design process the requirement of static stability has to be fulfilled. As soon as disturbances affect the aircraft during the flight, for example gust, a change in the flight attitude has to be corrected autonomously to the desired flight attitude. Aspiring a steady-state flight attitude requires a zero moment at the center of gravity [36]. Hence, the flight dynamic behavior is not only influenced by the magnitude of acting forces but also by the position of the center of gravity.

Table 4.11 shows the center of gravity of different prediction tools measured in x-direction of the aircraft. The position of the center of gravity refers to the global coordinate system, located at the nose of the aircraft. Thereby the discrete centers of gravity of the aircraft components are located in the plane of symmetry. The displayed data of the manufacturer are based on the documentation “A320 - Airplane Characteristics for Airport Planning” [11]. The results of the center of gravity show that the predicted magnitude of the Cöllen model in AC Builder is falsified by application of Torenbeek method from a deviation of -1% up to -3% regarding the magnitude as stated by the manufacturer. Nevertheless, Torenbeek method is applicable to predict the center of gravity well, but VAMPzero also attains accurate results.

The stability and control analysis in section 4.9 to 4.11 of the Cöllen model is based on the data set generated by *Weights and Balances* using Torenbeek method. Furthermore, the stand-alone tool *Weights and Balances* computes centers of gravity for a multitude of aircraft components as presented in Appendix C.

Table 4.11: Centers of gravity [m]

Model Method	Airbus -	Pester AC Builder	VAMPzero VAMPzero	Cöllen AC Builder	Cöllen Torenbeek
Wing	–	–	14, 48	–	15, 26
Horizontal tail	–	33, 81	34, 35	34, 45	35, 52
Vertical tail	–	34, 00	31, 50	33, 92	34, 67
Fuselage	–	16, 85	16, 02	16, 47	16, 47
Propulsion	–	14, 72	–	14, 15	15, 51
MZFW	16,49	-	-	-	15,90
Deviation					-4%
MTOW	16,56	14,64	15,38	16,34	16,01
Deviation		-12%	-7%	-1%	-3%

4.7 Results of AC Builder

In order to conduct analysis of aerodynamic and flight mechanic behavior in CEASIOM, data of the Cöllen model are employed. The Pester model is analyzed by Pester [6].

In figure 4.9 the shape of the Cöllen model is depicted within the environment of AC Builder. Regarding a comparison of the figures 4.9a, 4.9b, and 4.9c, it has to be considered that the linear perspective cannot be switched off in AC Builder. Figure 4.10a illustrates a gap between the fuselage and the vertical tail. This gap causes an error during the mesh generation in SUMO, but the error does not occur when computing closed and non-sharpe-edged surfaces of aircraft components. Even if the tail of the fuselage is not considered, a small gap between the fuselage and the vertical tail remains. While in VAMPzero the computed vertical location of the vertical tail is 2,08m, the diameter of the fuselage is 4,15m, so the gap measures at least 5mm. In order to close the gap between the vertical tail and the fuselage, the *vertical_locale* of the vertical tail is reduced from 0,5 to 0,3, as depicted in figure 4.10. The modified VAMPzero data are of special interest, since this data are applied for further stability and control analysis within the frame of this thesis.

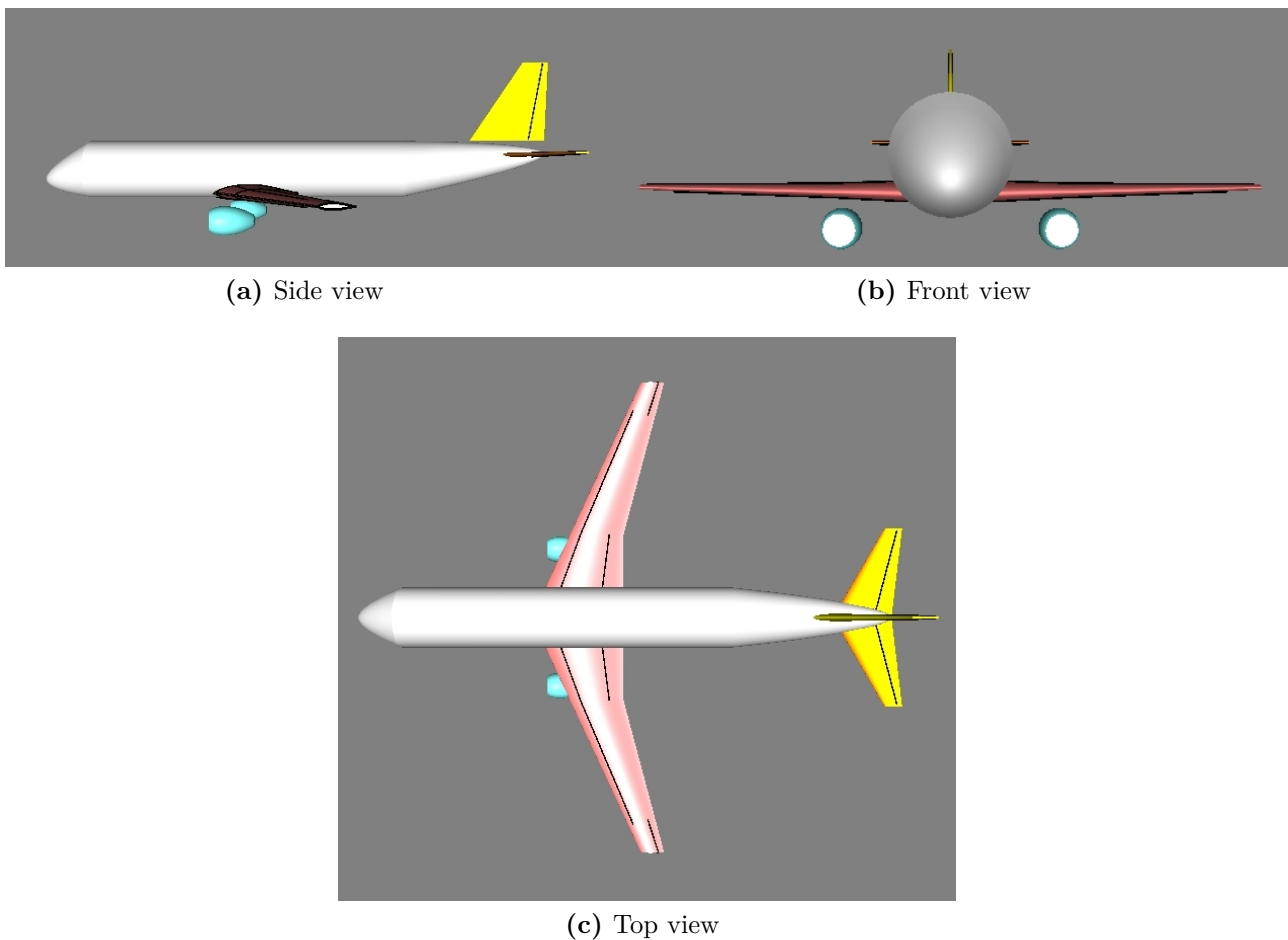


Figure 4.9: Cöllen model

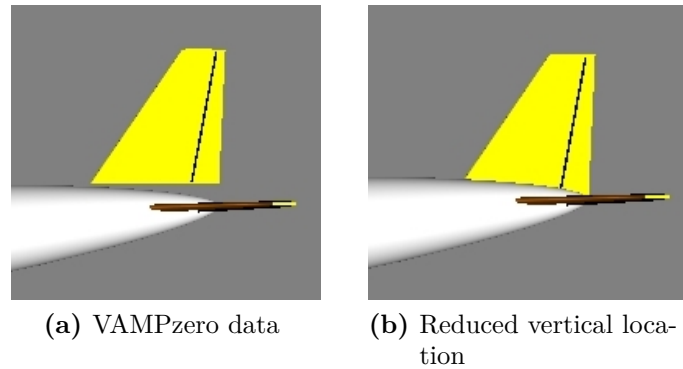


Figure 4.10: Modification of the vertical location of the vertical tail

4.8 Results of SUMO

In this part of the design process two meshes are generated in order to provide input data for CFD solutions based on Euler equations. At first a triangular surface mesh is created, following by the generation of a volume mesh using tetrahedral elements to fill the space between the aircraft and the farfield [21].

After import of the data set of *Weights and Balances* into SUMO the aircraft components listed in the GUI permit modifications by the user. The fuselage surface is modeled by cross sections, which are able to be modified interactively by editing parameters or by adjusting points of the fuselage skeleton. In figure 4.11 and 4.12 the fuselage skeleton is depicted in side and top view. The wing surface is created by airfoils, which are readout from the XML file and the CEASIOM airfoil library. The wing surface consists of linear or cubic b-splines [19]. Moreover, SUMO includes a control system editor to specify movable wing areas. Figure 4.13 shows the minimal and maximal deflection of the aileron, flaps, elevator, and rudder of the generated aircraft.

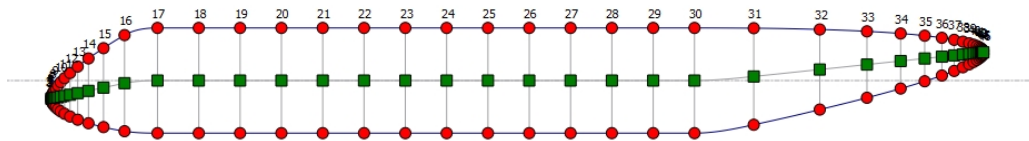


Figure 4.11: Skeleton of the fuselage, side view

The accuracy of CFD solutions is affected by the quality of the surface discretization. The surface mesh has to fulfill the following requirements:

- an accurate and sufficient approximation of the geometric surface
- a resolution of inviscid flow features such as pressure peaks and shocks

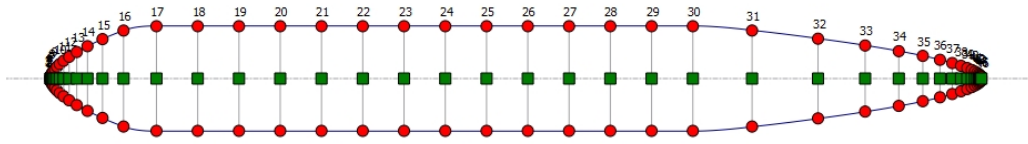


Figure 4.12: Skeleton of the fuselage, top view

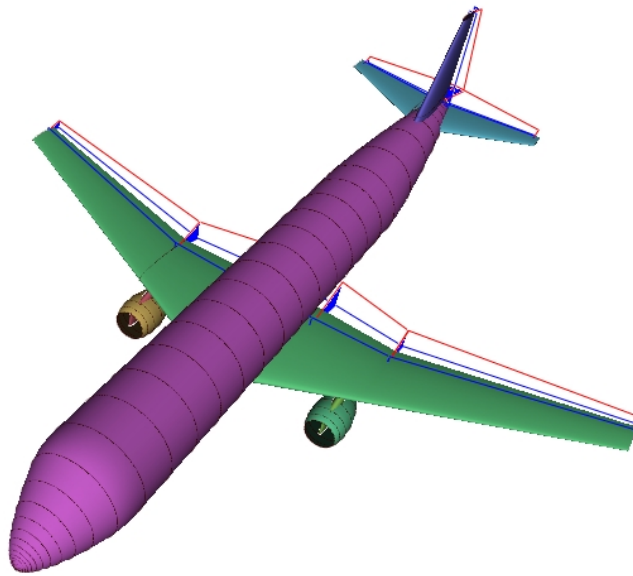


Figure 4.13: SUMO rendering of the A320

The first requirement is a necessary condition whereas the second condition is not always sufficient. To demonstrate this, the surface pressure distribution at the trailing edge of a lifting surface is considered. Even though the surface discretization in this area is flat, small elements are needed to indicate a high pressure gradient [27]. While the discrepancy could be solved by using solution-adaptive mesh refinement for a single flight condition, it is not a necessary option for a large quantity of cases. The algorithm implemented in SUMO generates a surface mesh, which satisfies the first requirement to given tolerances [21]. Local geometric quantities such as maximum stretch ratio, normal angle tolerance, and the number of iterations are used to define the discretization of the surface mesh. The following values are applied:

- maximum stretch ratio = 9
- normal angle tolerance = 15°
- number of iterations = 1

Furthermore, it is necessary to verify closed surfaces and non-sharp-edged transitions between the aircraft components. Otherwise, errors occur during the mesh generation process and the topology is not able to be closed.

The surface mesh of the A320 is displayed in figure 4.14 containing independent control surface definitions. Mesh regions that are affected by flap motion are marked as boundary condition regions in exported mesh files [19]. Table 4.12 implies specific results of the surface mesh.

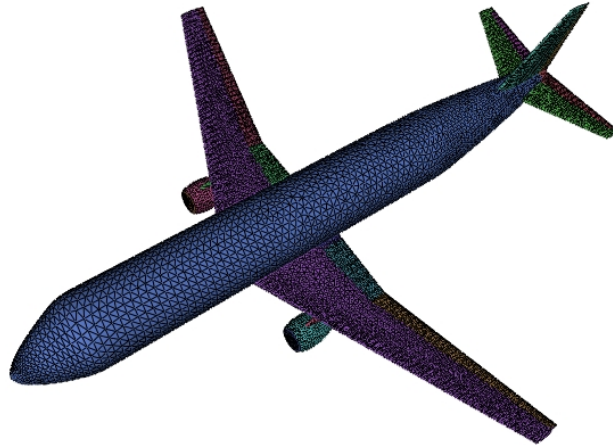


Figure 4.14: Surface mesh

Table 4.12: Surface mesh data

parameter	value
Triangles	69936
Vertices	34970
Wetted area	$751,093m^2$
Volume	$1347,27m^3$

The result of the unstructured volume mesh with refinement 4 is shown in figure 4.15. As long as the surface mesh has a sufficient quality the volume mesh is applicable for inviscid flow features. The volume mesh contains the geometric representation for Navier-Stokes computations in the EDGE CFD solver [27]. Table 4.13 displays the characteristics of four volume meshes required in AMB in section 4.9. Geometric quantities such as farfield radius, farfield refinement, and tet radius are specified whereas remaining parameters are computed in SUMO.

4.9 Results of AMB

In this part of the design process an aerodynamic table is generated to predict forces and moments for the stability and flight dynamic analysis. The aerodynamic table contains the following parameters

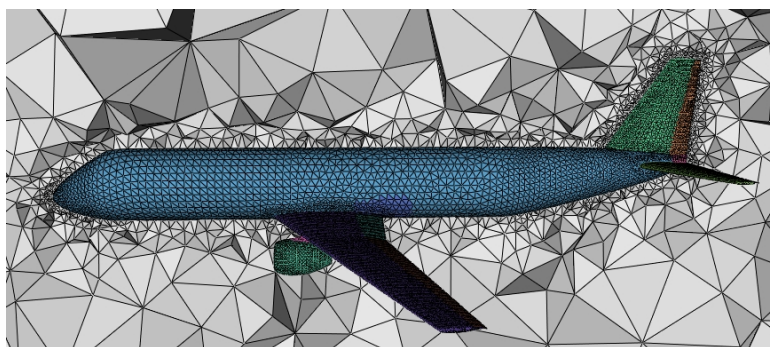


Figure 4.15: Volume mesh, refinement 4

Table 4.13: Volume mesh data

filename	Vol_mesh3	Vol_mesh4	Vol_mesh5	Vol_mesh6
Farfield radius	218, 12	218, 12	218, 12	218, 12
Farfield refinement	3	4	5	6
Tet radius/edge ratio	1, 4	1, 4	1, 4	1, 4
Max tet volume	5219	652	82, 0	10, 2
Nodes	111294	127683	261516	1345414
Boundary triangles	103128	106898	122312	183768
Tetrahedra	537380	635996	1472912	8355337

- α ,
- β ,
- Mach,
- δ_e , δ_a and δ_r ,
- Q, P, and R.

Providing these information the methods DATCOM, TORNADO, and EDGE Euler are applied. In this section the results of the different methods are compared to quantify the impact on the design sensitivity.

The input file of DATCOM and TORNADO is created by *Weights and Balances*. Hence, the data set includes, beside geometric information estimated weights, center of gravity, and moments of inertia. The EDGE Euler computation bases on a volume mesh with refinement 4 generated in SUMO. This mesh models the aircraft with sufficient accuracy. In contrast to finer volume meshes it requires a short calculation period. Specifications of necessary parameters and the results of the aerodynamic table are displayed in Appendix D. Figure 4.16 shows an extract of the results of GEO TORNADO providing the required data to execute TORNADO. Based on the panel allocation at the wing, and at the horizontal and vertical tailplane as depicted in figure 4.16a, the Vortex Lattice Method is applied. In figure 4.16b the positions of the reference

point and the center of gravity of the aircraft are illustrated. Due to the results of *Weights and Balances* both points are located within the range of the Mean Aerodynamic Chord (MAC).

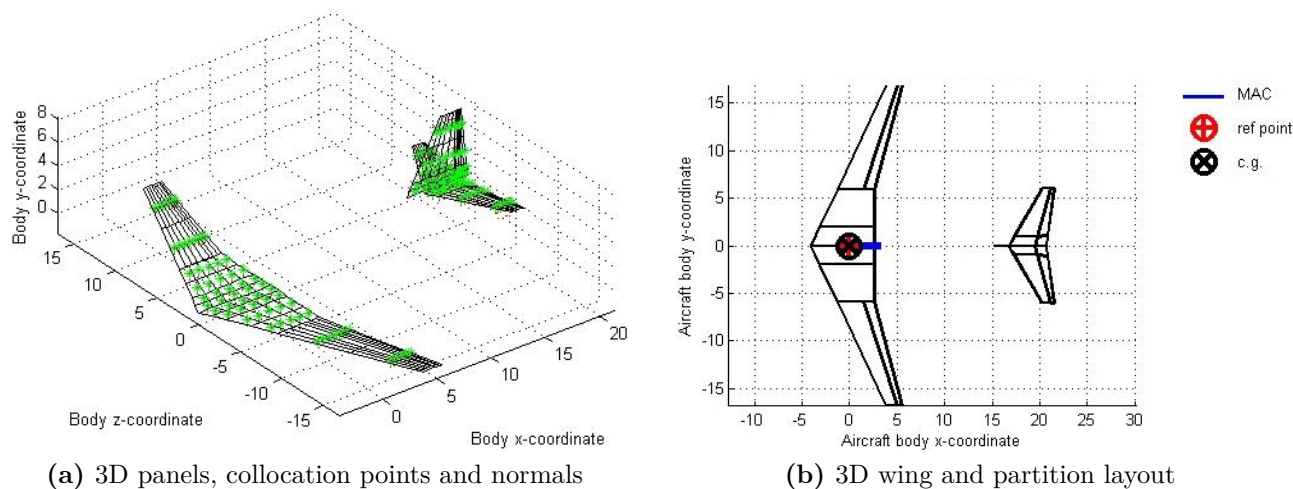


Figure 4.16: GEO TORNADO

The following figures show the aerodynamic coefficients C_L , C_D and C_m of the methods DATCOM, TORNADO, and EGDE Euler. The results are recorded for Mach = 0,6 and $\beta = 0^\circ$. Figure 4.18 illustrates the results of the Cöllén model in AMB containing the NACA airfoil 653218 of the main wing as depicted in figure 3.5b.

In order to quantify the design sensitivity of EDGE Euler method, six test cases are computed for different angles of attack within the range of -5° up to 15° . The aerodynamic coefficients C_L , C_D , and C_m are readout from the generated log file. However, the calculated aerodynamic coefficients C_L and C_D describes misleadingly the coefficients C_Z and C_X regarding the body-fixed coordinate system. Caused by variations of the angle of attack between the body-fixed and the aerodynamic coordinate system, the coefficients C_L and C_D have to be recalculated for each test case. Due to the geometric relationship exemplarily depicted in figure 4.17 for the drag coefficient, the aerodynamic coefficients are computed by equation (4.20) and (4.21). As long as $\beta = 0^\circ$ the y_f -axis of the body-fixed coordinate system agrees with the y_a -axis of the aerodynamic coordinate system. So the pitching moment coefficient does not need to be recalculated.

$$C_L = C_Z \cos \alpha - C_X \sin \alpha \quad (4.20)$$

$$C_D = C_Z \sin \alpha + C_X \cos \alpha \quad (4.21)$$

A reliable evaluation of the results of EDGE Euler is feasible as long as the fluctuation of the mean value is known. Hence, the convergence history of the aerodynamic coefficients has to be verified for each test case. The aerodynamic coefficients converge for 500 iterations with a deviation of less than one percent within the last 200 samplings with exception of the test case at $\alpha = 15^\circ$. To achieve a more significant result, this test case is rerun for 1500 iterations. The aerodynamic coefficients oscillate steadily around the mean values within the last 1000

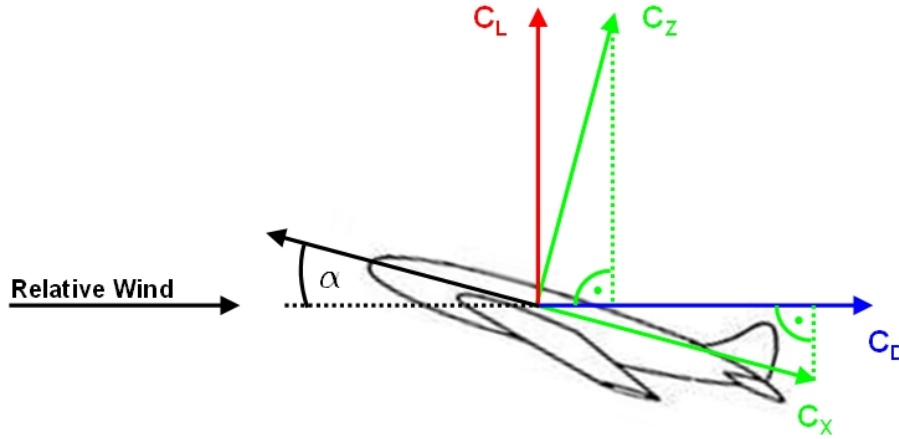


Figure 4.17: Composition of the drag coefficient

samplings as depicted in figure D.4. The identified deviations of the values are displayed in table 4.14 and visualized as error bars in figure 4.18. It has to be taken into account that the displayed drag coefficient in the log file does not coincide with the value of the last sampling in the bres file. This might be caused by different boundary flow conditions of the engines. The log file includes engines as closed bodies whereas the bres file considers the inlet of the engines, which considerably affects the drag. Therefore, the magnitude of the drag coefficient of the log file is used and the deviation of the mean value is readout of the bres file. However, it is recommended to exclude the engines in prospective analysis, not only in EDGE Euler but also in DATCOM and TORNADO, to avoid different interpretations.

Table 4.14: Deviation of aerodynamic coefficients at 1500 iterations

coefficient	C_L	C_D	C_m
deviation	8%	14%	13%

Figure 4.18a shows the lift coefficient against the angle of attack. The results of DATCOM and TORNADO coincide well with the approximation of EDGE Euler. Especially the results of the handbook method DATCOM are akin to the results of TORNADO. As long as the angle of attack is away from stall, TORNADO provides accurate results with the linear approximation of the lift coefficient. Stall occurs at $\alpha = 12^\circ$ and higher angles of attack visualized with EDGE Euler method.

Figure 4.18b shows the drag coefficient against the angle of attack. The almost constant deviation between DATCOM and TORNADO is caused by the impact of viscosity. In contrast to DATCOM, TORNADO does not consider the influence of viscosity on the drag. The impact of viscosity on the drag coefficient is described in equation (4.22) to (4.25) for subsonic cruise of a straight wing aircraft ($\Lambda_{SW} < 30^\circ$) by Raymer [37].

$$C_D = C_{Di} + C_{D0} \quad (4.22)$$

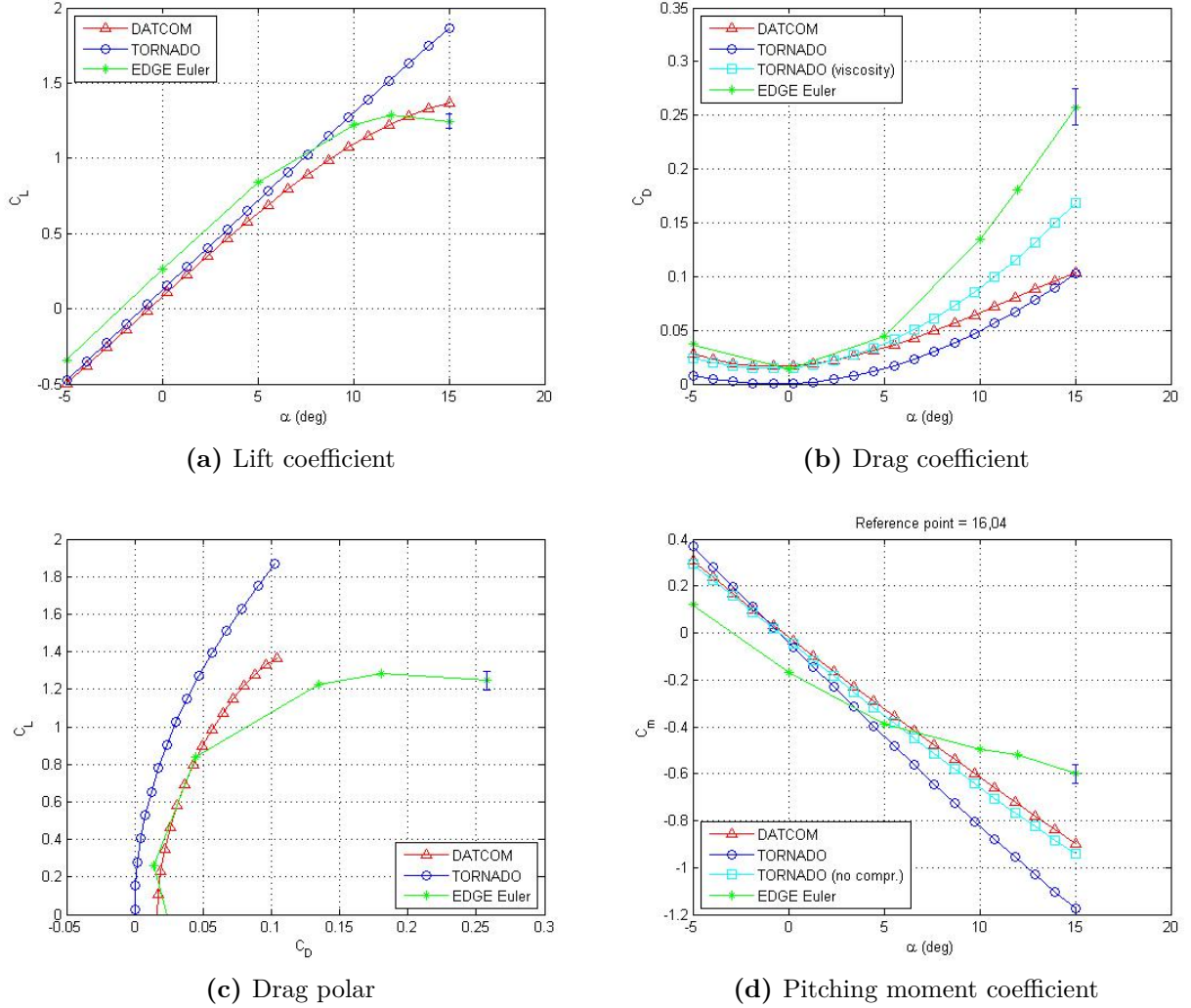


Figure 4.18: Aerodynamic coefficients, Mach = 0.6, $\beta = 0^\circ$

$$C_{Di} = \frac{C_L^2}{\pi \cdot e \cdot \Lambda} \quad (4.23)$$

$$C_{D0} = C_{fe} \cdot \frac{S_{wet}}{S_{ref}} \quad (4.24)$$

$$e = 1,78 \left(1 - 0,045\Lambda^{0,68}\right) - 0,64 \quad (4.25)$$

Including the parameters $\Lambda = 9,30$, $C_{fe} = 0,003$, and $\frac{S_{wet}}{S_{ref}} = 4,77$ the drag coefficient is sketched in figure 4.18b containing the influence of viscosity. In comparison to the previous result of TORNADO, the graph is shifted by a constant value and is stronger curved. The results of

DATCOM and TORNADO coincide well within the interval of -5° up to 5° for the angle of attack.

In figure 4.18c the drag polar is depicted as lift coefficient against the drag coefficient. Besides the load factor and altitude, the aerodynamic forces influences the flight time and range of the aircraft considerably. For that reason, this diagram is of special interest to detect the optimal lift and drag ratio so that the maximum range can be achieved [36]. The optimal point is located tangent to the line from the origin and closest to the vertical axis [37]. The deflection of the results between DATCOM and TORNADO is explained by the viscosity as in figure 4.18b.

Figure 4.18d displays the pitching moment coefficient against the angle of attack at reference point = 16,04m regarding the nose of the aircraft. The used reference point equates to the computed center of gravity by Torenbeek method displayed in table 4.11. The results of DATCOM, TORNADO, and EDGE Euler show a similar tendency with a negative gradient. Consequently, the neutral point is located behind the center of gravity and the aircraft possesses natural stability.

4.9.1 Conclusion

The low fidelity handbook method DATCOM generates aerodata for the stability and control analysis quickly, but predicts data for unconventional aircraft configurations inadequately. By means of the linear Vortex Lattice Method, TORNADO estimates aerodynamic solutions with sufficient accuracy for low speed and low angles of attack while the flow is reasonably attached. In contrast to DATCOM and TORNADO, the upgrade solver EDGE Euler deals with almost all configurations at all speeds before flow separation. Generally, EDGE Euler provides non-linear aerodynamics and predicts better results than TORNADO, though it is more time consuming. Table 4.15 shows the calculation period of DATCOM, TORNADO, and EDGE Euler for one test case. In TORNADO a quantity of 5 panels is chosen and for EDGE Euler 500 iterations are considered. The time periods are valid for a computer with windows environment, two cores with 1,67GHz and 2GB RAM.

Due to expensive EDGE Euler computations the TORNADO method is chosen for the stability and control analysis, which is sufficient to solve the attached flow problems for conventional aircraft configurations at low angles of attack. EDGE Euler method is recommended for configurations that are not able to be solved in TORNADO [26].

Table 4.15: Calculation period per test case

method	DATCOM	TORNADO	EDGE
time [sec/test case]	0,008	31	5400

4.10 Results of Propulsion

The Propulsion module generates the database for SDSA to permit stability and control analysis. The output of Propulsion is depicted in figure 4.19 showing the thrust of the engine against the Mach number according to the altitude of the aircraft.

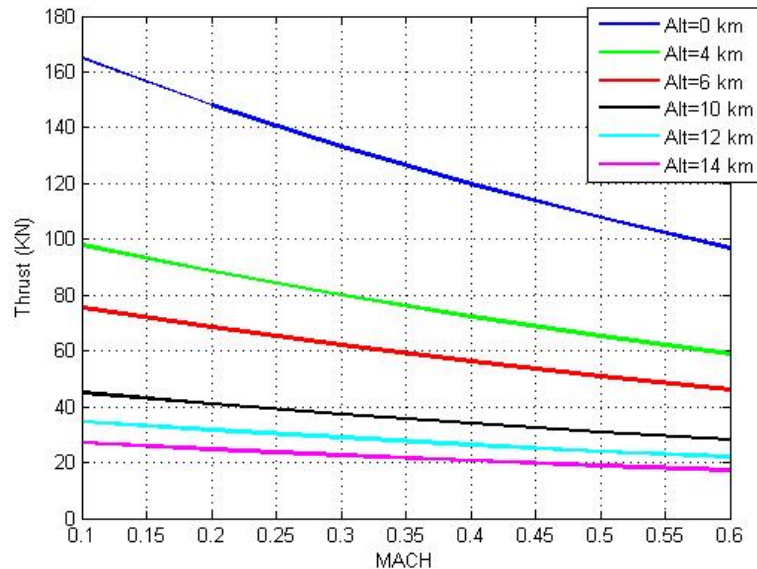


Figure 4.19: Propulsion output

4.11 Results of SDSA

The intention to generate the database of the flight envelope is not only to validate CFD data but also to predict stability and control properties by the SC module SDSA within CEASIOM [26]. Figure 4.20 shows the magnitudes of angle of attack and elevator deflection, which are trimming the Cöllén model for a range of flight speeds at various altitudes. Since the predictions are based on TORNADO-generated aerodynamic data up to a Mach number of 0,6, the results of SDSA are representable for corresponding low speeds and altitudes only. Hence, flight speeds within the range of 140m/s up to 220m/s and altitudes at 2000m up to 5000m are selected.

Stability and control design provide continuous authority over the aircraft within the flight envelope [26]. Figure 4.21 shows the stability criteria of the longitudinal motion predicted by SDSA. The longitudinal motion is subdivided into two distinct oscillations, a long-period oscillation designated as phugoid mode and a short-period oscillation referred to as short-period mode. The phugoid mode is a slow oscillation caused by the interchange of kinetic energy and

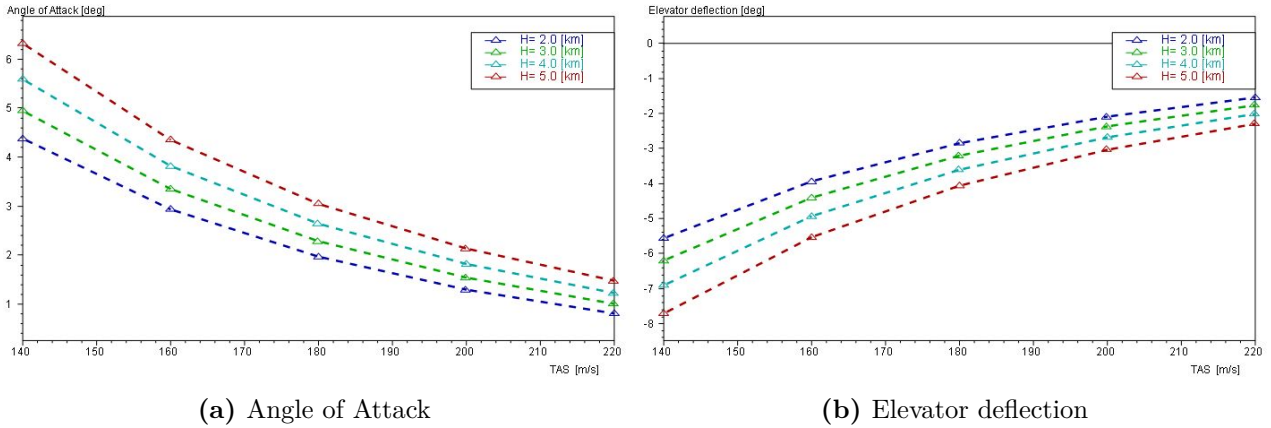


Figure 4.20: Trim results

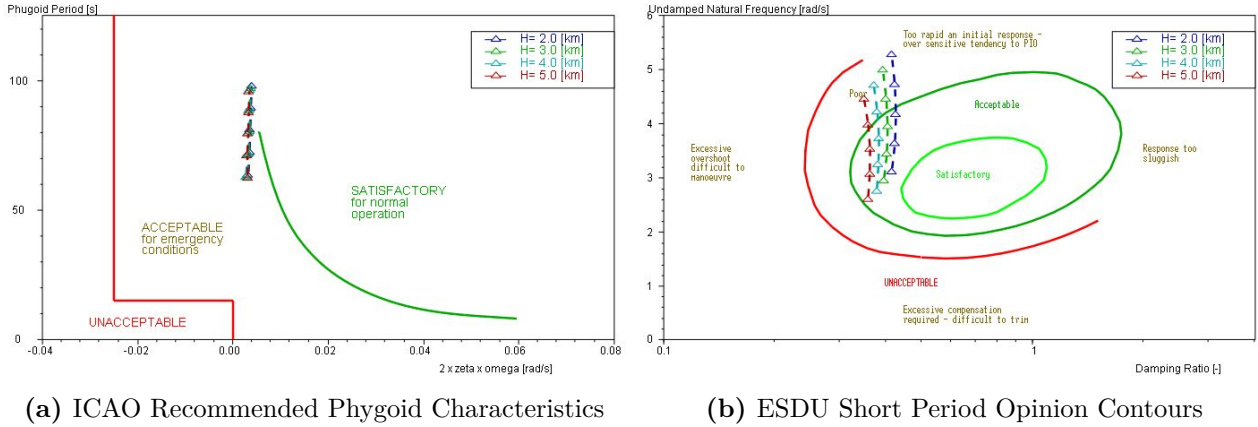


Figure 4.21: Stability criteria

potential energy. Figure 4.21a illustrates the phugoid period over the damping ratio times the frequency at different flight velocities and altitudes. By means of the relationship

$$\omega \approx \omega_0 = \frac{g}{v} \sqrt{2} \quad (4.26)$$

for an almost undamped oscillation, the angular frequency decreases for high air speeds and the phugoid period reduces with it. The short-period mode is a heavily damped oscillation leading to a fast dying wave motion of the aircraft. As depicted in figure 4.21b the short-period mode is shown as undamped natural frequency against the damping ratio. Since the eigen angular frequency ω_0 depends on the flight velocity, the angular frequency

$$\omega = \omega_0 \cdot \sqrt{1 - D^2} \sim v \quad (4.27)$$

rises with increasing flight velocity. Hence, the undamped natural frequency leaves the area *Acceptable* in figure 4.21b for high air speeds [36].

The aircraft control system should possess a characteristic response to avoid control reversal entirely. However, the calibration of the sensitivity of control displacements should not be too stable. Otherwise, the aircraft is not able to conduct maneuvers rapidly.

5 N2 Chart

The N2 Chart is a visual matrix representing functional or physical interfaces between system elements. The diagram is applied to identify, define, tabulate, design, and analyze functional and physical interfaces. The variable N denotes the number of elements in the $N \times N$ matrix whose relationships are illustrated. The user arranges the functional entities on the diagonal axis whereas the interface inputs are placed in the column and the outputs in the row of the function. A blank square constitutes no interface between the respective entities. Figure 5.1 depicts a directional flow of the interfaces between entities within the N2 Chart. Data flows in a clockwise direction between the functions. The symbol $F1 \rightarrow F2$ indicates the data flow from function F1 to function F2 and the symbol $F1 \leftarrow F2$ denotes the feedback. Passing data of the interfaces are defined in the applicable squares [10].

N2 diagrams are valuable tools to identify functional or physical interfaces and to pinpointing areas where conflicts might occur. In order to conclude this thesis the data flow information of used design tools are summarized in a N2 Chart in figure 5.2.

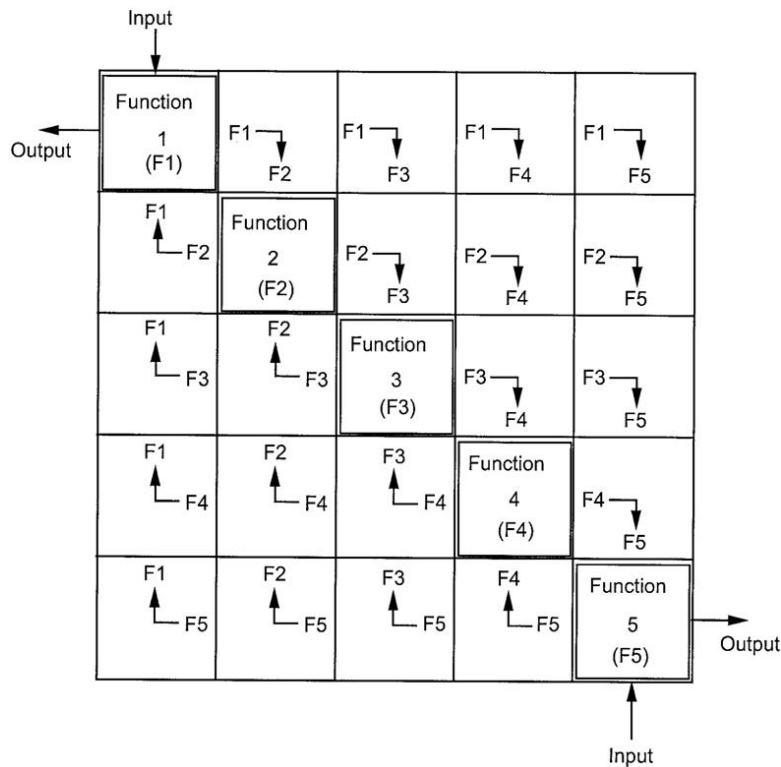


Figure 5.1: Composition of N2 Chart [10]

VAMPzero	<ul style="list-style-type: none"> →CPACS geometry → reference geometry → airfoils → weights → inertia, CoG → tool specific data → aeroPerformanceMap 							
	CPACS	<ul style="list-style-type: none"> →CPACS geometry →airfoils 						
	Wrapper	<ul style="list-style-type: none"> →CEASIOM geometry →airfoils →control surfaces (h) 	<ul style="list-style-type: none"> →CEASIOM geometry →airfoils →control surfaces (h) →fuselage a0,a1,b1 (m) →location wings (m) →Root_Airfoil etc.(m) →Nacelle1 , Nacelle2 (m) 	<ul style="list-style-type: none"> →CEASIOM geometry →airfoils →control surfaces (h) →weights (h) →inertia (h) →CoG (h) 				
			<ul style="list-style-type: none"> →reference geometry 	<ul style="list-style-type: none"> →CEASIOM geometry →airfoils →control surfaces (h) →fuselage a0,a1 b1 →location wings →Root_Airfoil etc.(m) →Nacelle1 , Nacelle2 	<ul style="list-style-type: none"> →CEASIOM geometry →airfoils →control surfaces (h) →weights (ACB) →inertia (ACB) →CoG (ACB) 			
			AC Builder					
				Weights and Balances				
						SUMO		
							→volume mesh	
								AMB

Figure 5.2: N2 Chart [h=hardcoded, m=manual inserted]

6 Conclusion and Suggestion for Future Projects

Within the framework of this thesis a wrapper is developed to transfer data from CPACS into CEASIOM. The wrapper is employed for a conventional transport aircraft with the Airbus A320 as reference configuration. Reference data of the manufacturer are compared with data of a novel aircraft designed by the conceptual aircraft design tool VAMPzero. Subsequently, the aerodynamic and flight mechanic behavior of the novel aircraft is analyzed within the CEASIOM components AMB, Propulsion, and SDSA. Using the aerodynamics methods DATCOM, TORNADO, and EDGE Euler allows to evaluate the impact on design sensitivities. In a final step the input and output correlations of the conceptual design tools are presented in a N2 Chart.

The development of the wrapper shows that precise information about the data structure of CPACS and CEASIOM components must be well-known, especially the minimum number of required parameters. This implies deep knowledge about parameter definitions in each design tool due to a multitude of coordinate systems. Parameters that are currently not provided by VAMPzero are hardcoded in the wrapper. A considerable challenge is to set up a wrapper with a flexible environment for various aircraft configurations.

A comparison between the Pester model and the Cöllen model shows that the geometric design is comparatively similar, mainly influenced by the input file of VAMPzero. Additionally, VAMPzero generates a weight breakdown, moments of inertia, and center of gravity with satisfactory results. In contrast to VAMPzero, the results of Cöllen (AC Builder) deviate stronger from the reference data of the manufacturer. Partly improved by the Torenbeek method of the stand-alone tool *Weights and Balances*, the weight predictions approach the reference magnitudes. However, the predictions of moments of inertia and center of gravity are slightly off the Torenbeek method.

The application of the aerodynamic methods DATCOM, TORNADO, and EDGE Euler identifies advantages and disadvantages of the different methods. By means of an adequate calculation period the TORNADO method is selected for further stability and control analysis. TORNADO is based on linear VLM generating accurate results for an conventional aircraft configuration with attached flow at low speed and low angle of attack. The handbook method DATCOM generates aerodata quickly, but with less accuracy than of TORNADO. The time-consuming and higher-fidelity EDGE Euler code is applicable for researches of unconventional configurations at all speeds before flow separation. The stability and control analysis in SDSA shows that the flight dynamic behavior of the Cöllen model (Torenbeek) is reviewed as „acceptable“ for operations at low speeds and altitudes.

The result of this thesis is a solid basis to adopt CPACS as a standardized interface into the framework of CEASIOM. Currently, CPACS is being implemented in AC Builder, which induces big efforts in rewriting the source code. Hence, AC Builder will be used as an editor to visualize CPACS data in CEASIOM. In order to close the design loop in conceptual design, the

wrapper has to be extended to transfer data from CEASIOM back to CPACS. As an example, the data generated in SDSA could be imported via CPACS into flight simulation tools. CPACS is an important step towards a unified data interface for aircraft design software. Furthermore, integration of further conceptual design tools such as RDS and PASS to CPACS might be a valuable advancement. The benefits of a standardized interface between various design tools will be experienced by all parties.

A Wrapper

Table A.1: Necessary parameters for AC Builder

component	parameter	value
baggage	installation_type	1
	gross_volume	90
cabin	Baggage_combined_length	0
	Baggage_apex_per_fuselgt	0
	Cabin_length_to_aft_cab	0
	Cabin_max_internal_height	0
	Cabin_max_internal_width	0
	Cabin_floor_width	0
	Cabin_volume	0
miscellaneous	Passenger_accommodation	150
	Design_classification	0

Table A.2: Calculated parameters in AC Builder

component	parameter
Fuselage	Nose_length
	Tail_length
	a0_fore, a1_fore, b1_fore
	a0_aft, a1_aft, b1_aft
Wing1	x, y, z
	AR
Horizontal_tail	x, y, z
	AR
Vertical_tail	x, y, z
	AR
Engines1	x, y, z
	x, y, z

Required parameters in SUMO:

Fuselage

- a0_fore, a1_fore, b1_fore, a0_aft, a1_aft, b1_aft

- a0_nose, a1_nose, b1_nose, a0_tail, a1_tail, b1_tail
- x, y, z

Wing

- Root_Airfoil, Kink1_Airfoil, Kink2_Airfoil, Tip_Airfoil
- x, y, z

Horizontal tail

- Root_Airfoil, Kink_Airfoil, Tip_Airfoil
- x, y, z

Vertical tail

- Root_Airfoil, Kink_Airfoil, Tip_Airfoil
- x, y, z

Without integration of the parameters x, y and z in the XML file the components of the aircraft are located at the origin of the global coordinate system.

Table A.3: Input parameters of VAMPzero

Parameter	Value
Aerodynamic	
Lift-over-drag ratio for loiter	11
Performance	
Cruise Altitude	12500 <i>m</i>
desRange	3200 <i>km</i>
Cruise Mach Number	0,78
Landing Field Length	1470 <i>m</i>
TO Field Length	2336 <i>m</i>
Propulsion	
nEngine	2
bypassRatio	5,9
Mass	
mPayload	19220 <i>kg</i>
paxSeats	150
Wing	
area	122,4 <i>m</i> ²
aspectRatio	9,3
taperRatio	0,25
phi25	24,967
xRoot	12 <i>m</i>
Htp	
area	31 <i>m</i> ²
taperRatio	0,303
Vtp	
area	21,5 <i>m</i> ²
Span	5,8 <i>m</i>
aspectRatio	1,5868
taperRatio	0,334
phi25	34

B AC Builder

Wing: Constant parameters

- airfoil = NACA653218
- reference_convention = 1
- configuration = 0
- winglet.present = 0
- fairing.present = 0

The data of the flap, aileron and slat are shown in figure B.1.

flap.present		<input checked="" type="checkbox"/>
flap.root_chord	[0-1]	0,275
flap.kink1_chord	[0-1]	0,275
flap.kink2_chord	[0-1]	0,275
aileron.present		<input checked="" type="checkbox"/>
aileron.chord	[0-1]	0,275
aileron.Span	m	0,213
aileron.position	0,1,2	1,0
aileron.limit_deflection_up	deg.	25,0
aileron.limit_deflection_d...	deg.	10,0
slat.present		<input checked="" type="checkbox"/>
slat.chord	[0-1]	0,22
slat.root_position	[0-1]	1,0
slat.tip_position	[0-1]	0,808

Figure B.1: Data of flap, aileron and slat

Horizontal tailplane: Constant parameters

- airfoil = NACA0000
- empennage_layout = 0
- limit_tailplane_deflection_up = 30°
- limit_tailplane_deflection_down = 15°

The data of the elevator is shown in figure B.2.

Vertical tailplane: Constant parameters

Elevator.present		<input checked="" type="checkbox"/>
Elevator.chord	[0-1]	0,32
Elevator.Span	[0-1]	0,95
Elevator.limit_deflection_up	deg.	30,0
Elevator.limit_deflection_...	deg.	15,0

Figure B.2: Elevator data

- airfoil = NACA0000
- Twin_tail = 0

The data of the rudder is shown in figure B.3.

Rudder.present		<input checked="" type="checkbox"/>
Rudder.chord	[0-1]	0,23
Rudder.Span	[0-1]	0,95
Rudder.limit_deflection	deg.	30,0

Figure B.3: Rudder data

Engine: Constant parameters

- Layout_and_config = 0
- Propulsion_type = 0
- Nacelle_body_type = 1
- symmetry = 1

C Weights and Balances

Additional input parameters

- Range = 3200km
- Avionic component mass = 0kg
- Main landing gear length = 1,07m
- Nose landing gear length = 0,94m

The magnitudes of the lengths of the main landing gear and nose landing gear are based on the documentation of “A320 - Airplane Characteristics for Airport Planning” by Airbus [11].

The additional input parameters are required for the weight estimation methods Torenbeek, Raymer, USAF, and Cessna [31].

Constant values

- $A1 = 0,55$
- $B1 = 0,72$
- $C5 = 1,24$
- $Kf = 0,81$
- $Klg = 0,035$
- $Kpr = 2$
- $Kfs = 0,011$
- $Kfc = 0,01$
- $Kls = 0,008$
- $Kav = 0,007$
- $Kfur = 40$

A description of these parameters is given in the documentation of Denis Howe [38]. The magnitudes of the parameters are standardized implemented in *Weights and Balances*. Little changes in the constant values might effect large variations on the weight breakdown.

Weight breakdown of different estimation methods

Centers of gravity: Result of *Weights and Balances*

C Weights and Balances

	Howe	Torenbeek	Raymer	USAF	Cessna
Wing 1	10266.06	9117.57	3991.93	4065.87	2032.41
Wing2	0.00	0.00	0.00	0.00	0.00
Horizontal Tail	623.70	665.25	552.11	440.15	873.95
Vertical Tail	415.73	418.76	2429.87	158.05	2664.07
Fuselage	219308.04	4213.14	3664.47	6553.62	5317.84
Landing Gear	10536.38	2520.99	1426.95	1571.94	2004.28
Canard	0.00	0.00	0.00	0.00	0.00
Tail booms	0.00	0.00	0.00	0.00	0.00
Total Structure	241149.91	16945.86	12065.32	12789.63	12919.51
Total propulsion system	12571.17	7238.58	8119.94	8119.94	9280.04
Fuel system	3311.43	198.90	610.80	610.80	1004.00
Flight Control	3010.39	642.00	3526.86	734.75	1063.69
Avionic	2107.28	928.87	0.00	0.00	
Electric system	2408.32	0.47	1162.67	224.86	1696.84
Air conditioning	320.00	694.44			
Hydraulic system			59.05		
Furnishing	6000.00	7500.00	3407.47	2221.62	19011.59
Total system	17157.42	9964.66	8766.85	3792.03	22776.13
Total empty weight	270878.50	34138.95	28952.12	24701.60	44962.21
Payload	14850.00	14850.00	14850.00	14850.00	14850.00
Fuel	15311.00	15311.00	15311.00	15311.00	15311.00

Figure C.1: Weight breakdown of *Weights and Balances*

	Weight	X	Y	Z
Wing1	9117.57	15.26	0.00	-1.
Wing2	0.00			
Horizontal Tail	665.25	35.52	0.00	1.
Vertical Tail	418.76	34.67	0.00	3.
Fuselage	4213.14	16.47	0.00	-0.
Canard	0.00			
Tailbooms	0.00			
Engine1	7238.58	15.51	5.06	-2.
Engine2	0.00			
Passengers	14850.00	14.80	0.00	-1.
Fuel tank in wing1	15311.00	16.62	0.00	-1.
Fuel tank in wing 2	0.00			
Fuel auxiliary	0.00			
Center of Gravity at Empty Weight		16.5744	0	-1.6964
Center of Gravity at Zero Fuel Weight		15.8509	0	-1.4509
Center of Gravity at Take Off Weight		16.0582	0	-1.4250
b _x	1695313.69			
b _{yy}	2160994.47			
b _{zz}	3712331.35			
b _z	79854.73			
b _y	0.00			
b _{yz}	0.00			

Figure C.2: Centers of gravity

D AMB

Specifications of parameters

- Flight Dynamics Model = 6 DOF
- Geometry Symmetry Plane = yes
- Wing airfoil = NACA653218
- Horizontal tail airfoil = NACA0000
- Vertical tail airfoil = NACA0000

Parameter	Units	Value
Wing Area	m ²	122.3999
Long. Ref. Length	m	3.8281
Lat. Ref. Length	m	33.7390
XCG From Nose	m	16.0068
YCG	m	0
ZCG Fus. Centerline	m	-1.1790
X 1/4 MAC from Nose	m	16.2069
Maximum TO Weight	kg	6.4300e+04
Ixx	kg m ²	2.2378e+06
Iyy	kg m ²	2.1673e+06
Izz	kg m ²	4.2579e+06
Ixz	kg m ²	8.2168e+04
Ixy	kg m ²	0
Iyz	kg m ²	0

Figure D.1: Reference data

Parameter	Units	Min Value	Max Value	N Values
AoA	Deg	-5	15	5
Beta	Deg	-5	5	2
Mach		0.1000	0.6000	3
Elev.	Deg	-15	30	2
Ail	Deg	-10	25	2
Rud	Deg	-30	30	2
Pitch Rate	Deg/s	-10	10	2
Roll Rate	Deg/s	-10	10	2
Yaw Rate	Deg/s	-10	10	2

Figure D.2: States

Required information for DATCOM

Type of flap

Parameter					
AoA	-5	0	5	10	15
Beta	-5	5			
Mach	0.1000	0.3500	0.6000		
Ele.	-15	30			
Ail.	-10	25			
Rud.	-30	30			
Pitch Rate	-10	10			
Roll Rate	-10	10			
Yaw Rate	-10	10			

Figure D.3: Aerodynamic table

- Elevator = Plain Flap
- Rudder = Plain Flap

Required information for TORNADO

- Number of Panels in x direction = 5
- Number of Panels in y direction = 5
- Number of Panels for flapes = 5
- Compressibility Effects = 1
- Viscous Effects = 0
- Analysis = Brute-Force Calculation

Required information for EDGE Euler

- Altitude = 12500m
- XML/Grid Scaling Factor = 1
- Number of Iterations = 500
- Multigrid (1-Yes, 0-No) = 1
- Number of Multigrid Levels = 4
- Sampling Tolerance = 1,0000e-05
- Maximum Number of Samples = 50
- Mach Number = 0,6
- $\beta = 0^\circ$
- Control Surf Analysis = disabled

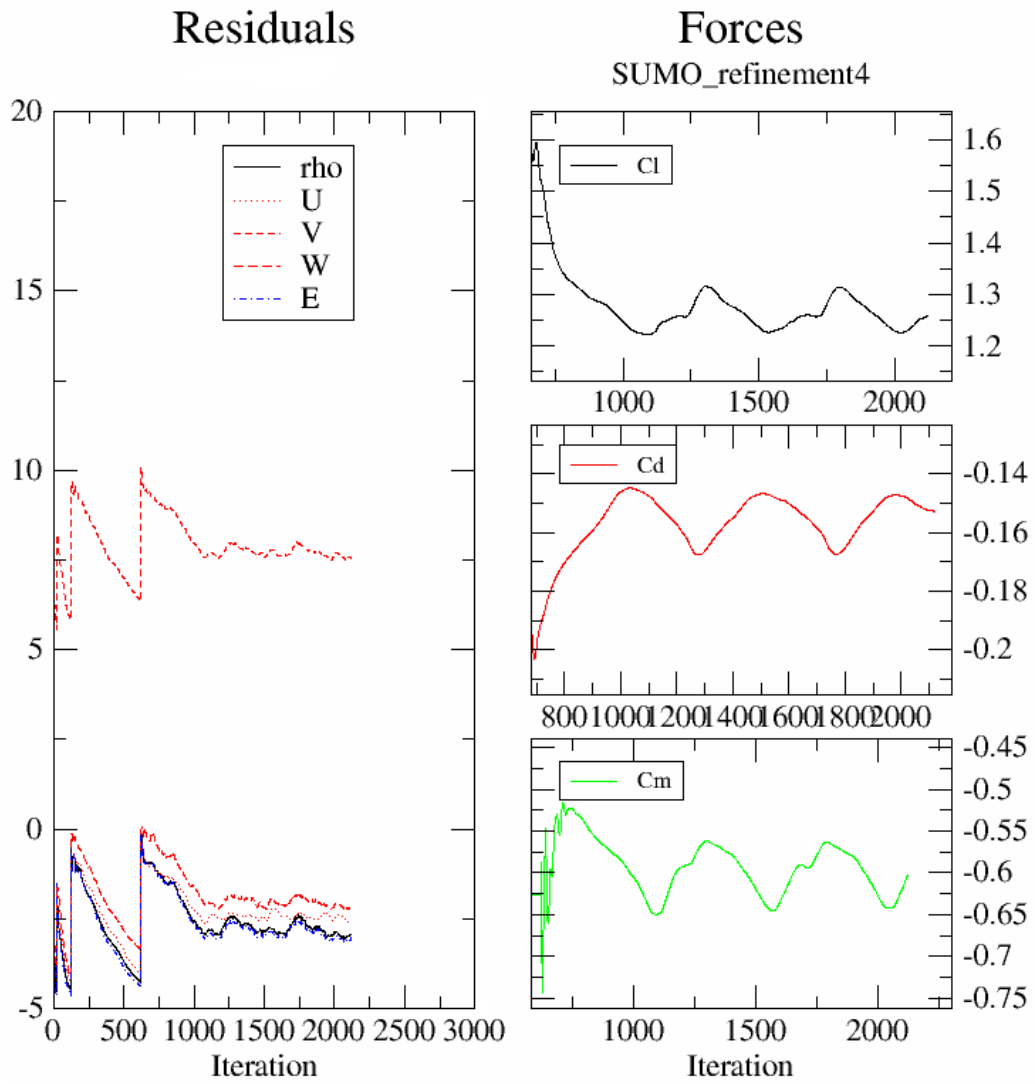


Figure D.4: EDGE Euler convergence history, Mach = 0.6, $\alpha = 15^\circ$, $\beta = 0^\circ$

Bibliography

- [1] PFEIFFER, Till: *Implementation of a Heterogeneous, Variable-Fidelity Framework for Flight Mechanics Analysis in Preliminary Aircraft Design*. Hamburg University of Technology, Institute of Air Transportation Systems, German Aerospace Center, Hamburg, 2011
- [2] BÖHNKE, Daniel: *An Approach to Multi-fidelity in Conceptual Aircraft Design in Distributed Design Environments*. Hamburg University of Technology, Institute of Air Transportation Systems, German Aerospace Center, Hamburg, 2011
- [3] SIMSAC: *CEASIOM modules*, 2011. <http://www.ceasiom.com/ceasiom-modules.html>, Abruf: 01. Aug. 2011
- [4] PUELLES, Andrés: *CEASIOM XML file Definition*. 2010-02-17, February 2010. <http://www.ceasiom.com>, Abruf: 21th June 2011
- [5] VON KAENEL R., J. OPPELSTRUP, GOETZENDORF-GRABOWSKI T., GHOREYSHI M., CAVAGNA L., BERARD A.: *CEASIOM: Simulating Stability & Control with CFD/CFM in Aircraft Conceptual Design*. International Congress of the Aeronautical Sciences, 2008
- [6] PESTER, Maria: *Multi-Disciplinary Conceptual Aircraft Design using CEASIOM*. Master thesis. Hochschule für Angewandte Wissenschaften Hamburg, Department Fahrzeugtechnik und Flugzeugbau, Hamburg, 2010
- [7] GHOREYSHI M., D. VALLESPIN, A. DA RONCH, K. J. BADCOCK: *Simulation of Aircraft Manoeuvres Based On Computational Fluid Dynamics*. AIAA Aerospace Sciences Meeting, 2010
- [8] SIMSAC: *SDSA - Coordinates, signs and units definition*. SimSac Sixth Framework Programme, 2009
- [9] AEROSPACEWEB: *Airbus A320*, 2011. http://www.aerospaceweb.org/aircraft/jetliner/a320/a320_schem_01.jpg, Abruf: 29. Aug. 2011
- [10] NAS SYSTEM ENGINEERING MANUAL: *Section 4.4 - N-Squared Diagramming*. version 3.1, 2006
- [11] AIRBUS: *A320 - Airplane Characteristics for Airport Planning*. Airbus S.A.S., France, 2005
- [12] RAYMER, Daniel P.: *Aircraft Design: A Conceptual Approach*. Second Edition, AIAA Education Series, 1992
- [13] GHOREYSHI M., A. DA RONCH, K. J. BADCOCK, J. DEES,: *Aerodynamic Modelling for Flight Dynamics Analysis of Conceptual Aircraft Designs*. AIAA20094121. AIAA Applied Aerodynamics Conference, San Antonio, 2009

- [14] AIRBUS : *A320 Family - A320neo: Setting New Standards*, 2011. <http://www.airbus.com/aircraftfamilies/passengeraircraft/a320family/>, Abruf: 22. Aug. 2011
- [15] JEPSEN, Jonas: *Implementierung eines Geometriekonverters zur Übersetzung von Flugzeugparametrisierungen*. Thesis IB-328-2010-15. Hamburg University of Technology, Institute of Air Transportation Systems, Hamburg, 2010
- [16] DA RONCH A. , C. MCFARLANE, C. BEAVERSTOCK, J. OPPELSTRUP, M. ZHANG, A. RIZZI, JAVIER MUNOZ: *Benchmarking CEASIOM Software to Predict Flight Control and Flying Qualities of the B-747*. International Congress of the Aeronautical Sciences, 2010
- [17] ISIKVEREN, Askin: *Quasi-Analytical Modelling and Optimization Techniques for Transport Aircraft Design*. Report 2002-13. Royal Institute of Technology (KTH), Department of Aerodynamics, Sweden, 2002
- [18] MARCHANT, Ben: *The AC Builder Geometry Definition for CEASIOM I: Lifting Surfaces*. Project Report. Royal Institute of Technology (KTH), Aeronautical and Vehicle Engineering Department, Sweden, 2011
- [19] LAROSTERNA ENGINEERING DYNAMICS LAB: *Aircraft Modeling & Mesh Generation*, 2009. <http://www.larosterna.com/sumo.html>, Abruf: 18. Jul. 2011
- [20] SI, Hang: *On Refinement of Constrained Delaunay Tetrahedralizations*. Weierstrass Institute for Applied Analysis and Stochastics, Berlin, 2006
- [21] TOMAC, Maximilian ; ELLER, David: *From Geometry to CFD-Based Aerodynamic Derivatives - An Automated Approach*. International Congress of the Aeronautical Sciences. 2010
- [22] ELIASSON, Peter: *EDGE - A Navier-Stokes Solver for Unstructured Grids*. Report FOI-R-0298-SE. FOI Swedish Defence Research Agency, Stockholm, 2001
- [23] RIZZI, Arthur: *Modeling and Simulating Aircraft Stability & Control the SimSAC Project*. AIAA20108238. AIAA Guidance, Navigation and Control Conference, Toronto, Ontario, 2010
- [24] ANDERSON, J. D.: *Modern Compressible Flow*. McGraw Hill, 1990
- [25] MEADOR, William ; SMART, Michael: *Reference Enthalpy Method Developed from Solutions of the Boundary-Layer Equations*. NASA Langley Research Center. Hampton, Virginia, 2005
- [26] ZHANG, Mengmeng: *Application and Development of the CEASIOM-SUMO-EDGE Suite for Rapid AeroData Assessment of Aircraft Flying Qualities*. Licentiate thesis. Royal Institute of Technology (KTH), Aeronautical and Vehicle Engineering Department, Sweden, 2010
- [27] RIZZI A., J. OPPELSTRUP, M. ZHANG, M. TOMAC: *Coupling Parametric Aircraft Lofting to CFD & CSM Grid Generation for Conceptual Design*. AIAA Aerospace Sciences Meeting, 2011
- [28] SIMSAC: *SDSA - Theoretical Basics*. SimSac Sixth Framework Programme, 2009

- [29] REY, Benoit: *SimSac - CEASIOM Tutorial*. SimSac Sixth Framework Programme, 2010
- [30] ENGINE YEARBOOK: *Directory of Major Commercial Aircraft Turbofans*, 2007
- [31] MARTIN, Javier M.: *Further Development and Testing of CEASIOM for Weight, Balance and Structural Design of Aircraft*. Master thesis. Royal Institute of Technology (KTH), Aeronautical and Vehicle Engineering Department, Sweden, 2010
- [32] DORBATH, Felix: *LTH Mass Analysis: Civil Jet Transport Aircraft - Statistical Mass Estimation*. Report UL-442.0(T). DLR-LY and Airbus, Hamburg, 2011
- [33] GOLLNICK, Volker: *Methoden des Flugzeugentwurfs I*. Lecture document. Hamburg University of Technology, Institute of Air Transportation Systems, Hamburg, 2010
- [34] TORENBEEK, Egbert: *Synthesis of Subsonic Airplane Design*. Delft University Press, Netherlands, 1982
- [35] SCOTT, Paul W. ; NGUYEN, Dennis: *The Initial Weight Estimate*. SAWE Paper No. 2327. 55th International Conference of the Society of Allied Weight Engineers, Atlanta, 1996
- [36] HAHN, Ulrich: *Flugmechanik*. Lecture document. Hamburg University of Technology, Hamburg, 2010
- [37] RAYMER, Daniel P.: *Conceptual Design Modeling in the RDS-Professional Aircraft Design Software*. AIAA Paper No. 2011-161, 2011
- [38] HOWE, Denis: *Aircraft Conceptual Design Synthesis*. Cranfield, United Kingdom, 2000