GARTEUR GROUP FOR AERONAUTICAL RESEARCH AND TECHNOLOGY IN EUROPE

ORIGINAL: ENGLISH GARTEUR/TP-088-3

February 17, 1997

GARTEUR Open

Robust Flight Control Design Challenge
Problem Formulation and Manual:
the Research Civil Aircraft Model (RCAM)

by

FM(AG08)

GARTEUR aims at stimulating and co-ordinating
co-operation between Research Establishments and Industry
in the areas of Aerodynamics, Flight Mechanics, Helicopters,
Structures & Materials and Propulsion Technology

GARTEUR GROUP FOR AERONAUTICAL RESEARCH AND TECHNOLOGY IN EUROPE

ORIGINAL: ENGLISH GARTEUR/TP-088-3

February 17, 1997

GARTEUR Open

Robust Flight Control Design Challenge Problem Formulation and Manual: the Research Civil Aircraft Model (RCAM)

by

FM(AG08)

This report has been prepared under auspices of the Responsables for Flight Mechanics, Systems and Integration of the Group for Aeronautical Research and Technology in EURope (GARTEUR)

Group of Resp.	:	FM-GoR	Action Group	:	FM(AG08)
Report Resp.	:	A. Helmersson/	Version	:	3
Project Man.	:	J.C. Terlouw/	Completed	:	February 17, 1997
Monitoring Resp.	:	J.T.M. van Doorn/			

List of authors

Main authors:

Paul Lambrechts[1],
Samir Bennani[2],
Gertjan Looye[2],
Anders Helmersson[3].

Contributing authors:

Pierre Fabre, Joseph Irvoas, Philippe Ménard[4],
Jean-François Magni[5],
Alberto Martínez[6],
Dieter Moormann[7],
Stefano Scala[8],
Jan Schuring[1],
Phillip Sheen[9],
Jan Terlouw[1],
Hans van der Vaart[2].

[1] NLR, Anthony Fokkerweg 2, 1059 CM Amsterdam, The Netherlands.
E-mail lambo@nlr.nl, fax +31 20 5113210.
E-mail schuring@nlr.nl, fax +31 20 5113210.
E-mail terlouw@nlr.nl, fax +31 20 5113210.
[2] TUD, Kluyverweg 1, 2629 HS Delft, The Netherlands.
E-mail s.bennani@lr.tudelft.nl, fax +31 15 2786480.
E-mail hans.vandervaart@lr.tudelft.nl, fax +31 15 2786480.
[3] LiTH, Linköping University, S-581 83 Linköping, Sweden.
E-mail andersh@isy.liu.se, fax +46 13 282622.
[4] Aérospatiale, 316 Route de Bayonne, 31060 Toulouse Cedex 03, France.
E-mail philippe.menard@avions.aerospatiale.fr, fax +33 619 380 90.
[5] CERT-ONERA, 2 Av. Edouard Belin, 31055 Toulouse Cedex, France.
E-mail magni@jupiter.cert.fr, fax +33 622 525 64.
[6] INTA, Carretera de Ajalvir Km. 4, 28850 Torrejón de Ardoz, Madrid, Spain.
E-mail mtezba@inta.es, fax +34 1 520 1724.
[7] DLR, Postfach 1116, D-82230 Wessling, Germany.
E-mail dieter.moormann@dlr.de, fax +49 8153 28 1441.
[8] CIRA, Via Maiorise, 81043 Capua, Italy.
E-mail scst229@cira.it, fax +39 823 623 335.
[9] AVRO, Chester Road, Woodford, Stockport, Cheshire SK7 1QR, United Kingdom.
E-mail phillip.p.j.sheen@woodford.bae.eurokom.ie, fax +44 161 955 3085.

Summary

This document defines one of the two Robust Flight Control design challenges that are prepared by the GARTEUR Action Group FM(AG08). This design challenge is based on the Research Civil Aircraft Model (RCAM) and considers a civil aircraft during final approach.

A non-linear model is supplied to be used for design and simulation. The problem formulation is intended to be 'non-trivial' for flight control system design: a realistic set of design specifications is formulated in which a number of important trade-offs are to be considered. On the other hand, it may not be expected that these specifications are complete for actual flight control system design. A fully automated evaluation procedure is developed to be able to compare the performance of resulting controllers, irrespective of the methods used to design them.

It is expected from design challenge participants that they supply insight into the usefulness of the considered method(s), not only in the sense of controller performance, but also in the sense of controller complexity and controller design time. Especially the latter is considered to be most important for the aircraft industry.

Distribution list

FM(AG08) Principal Persons

Ahmed S (UK)	CCL	1 copy
Ambrosino G (IT)	UNAP	1 copy
Bernussou J (FR)	LAAS	1 copy
Cruz J de la (ES)	UCM	1 copy
Delgado I (ES)	INTA	1 copy
Dormido S (ES)	UNED	1 copy
Duda H (DE)	DLR	1 copy
Escande B (FR)	ONERA	1 copy
Faleiro L (UK)	LUT	1 copy
Fielding C (UK)	BAe-MA	1 copy
Game G (UK)	BAeDD	1 copy
Gautrey J (UK)	CUN	1 copy
Helmersson A (SE)	LiTH	1 copy
Joos H (DE)	DLR	1 copy
Laidlaw D (UK)	AVRO	1 copy
Luckner R (DE)	DASA	1 copy
Maciejowski J (UK)	UCAM	1 copy
Magni J (FR)	ONERA	1 copy
Muir E (UK)	DRA	1 copy
Postlethwaite I (UK)	ULES	1 copy
Schram G (NL)	DUT-EE	1 copy
Ståhl-Gunnarsson K (SE)	SMA	1 copy
Terlouw J (NL)	NLR	1 copy
Tonon A (IT)	ALN	1 copy
Vaart J van der (NL)	DUT-AE	1 copy
Verde L (IT)	CIRA	1 copy

Members of the Flight Mechanics, Systems and Integration Group of Responsables

Brännström B (SE)	FMV	1 copy
Doorn J van (NL)	NLR	1 copy
England P (UK)	DRA	1 copy
Muñoz F (IT)	INTA	1 copy
Rodloff R (DE)	DLR	1 copy
Verbrugge R (FR)	ONERA	1 copy

Members of the Executive Committee

Chéret J (FR)	ONERA	1 copy
Coleman G (UK)	DRA	1 copy
Gustafsson A (SE)	FFA	1 copy
Haupt R (DE)	DLR/PD-L	1 copy
Moreno Labata G (ES)	INTA	1 copy
Slooff J (NL)	NLR	1 copy

Secretary GARTEUR

Jonasson K (SE)	FFA	1 copy

Contents

List of figures

List of tables

List of symbols and abbreviations

Symbols

The used symbols are according to the nomenclature defined in the Communication Handbook [2].

Abbreviations

Organisations/Countries:

ALN	Alenia Aeronautica
AVRO	Avro International Aerospace
BAeDD	MATRA - British Aerospace Defence Dynamics
BAe-MA	British Aerospace Military Aircraft
CCL	Cambridge Control
CERT	Centre d'Etudes et de Recherches de Toulouse
CIRA	Centro Italiano Ricerche Aerospaziali
CUN	Cranfield University
DASA	Daimler-Benz Aerospace Airbus
DE	Germany
DLR	Deutsche Forschungsanstalt für Luft- und Raumfahrt
DRA	Defence Research Agency
DUT-AE	Delft University of Technology, Department of Aerospace Engineering
DUT-EE	Delft University of Technology, Department of Electrical Engineering
ES	Spain
FFA	The Aeronautical Research Institute of Sweden
FMAG	Flight Mechanics Action Group
FM-GoR	Flight Mechanics, Systems and Integration Group of Responsables
FMV	Defense Material Administration
FR	France
GARTEUR	Group for Aeronautical Research and Technology in EURope
INTA	Instituto Nacional de Técnica Aerospacial
IT	Italy
LAAS	Laboratoire d'Analyse et d'Architecture des Systemes
LiTH	Linköping University
LUT	Loughborough University
NL	The Netherlands
NLR	National Aerospace Laboratory

ONERA	Office National d'Etudes et de Recherches Aérospatiales
SE	Sweden
SMA	Saab AB, Saab Military Aircraft
UCAM	University of Cambridge
UCM	Universidad Complutense Madrid
UK	United Kingdom
ULES	University of Leicester
UNAP	Universitá degli Studi di Napoli Federico II
UNED	Universidad Nacional de Educación a Distancia

Other:

AC	Aerodynamic Centre
ANDECS	ANalysis and DEsign of Controlled Systems
CoG	Centre of Gravity
RCAM	Research Civil Aircraft Model

1 Introduction

1.1 Objectives of GARTEUR Action Group FM(AG08)

In this document a Robust Flight Control design benchmark problem is proposed. It has been prepared by GARTEUR Action Group FM(AG08) on "Robust Flight Control (RFC) in a Computational Aircraft Control Engineering Environment (CACEE)". The objectives and activities of this Action Group will be discussed in the following. More details can be found in the FM(AG08) Terms of Reference [1].

A theme of world-wide importance to aircraft manufacturing companies is the improvement and automatisation of techniques for computer-aided aircraft design integration, which goes beyond mere functional integration of aircraft components and seeks to provide optimal performance for the vehicle as a whole. In [12] the following observation is made:

> The traditional process of systems integration is to make individually designed subsystems work together on an aircraft, that is, to ensure compatibility and minimise adverse interactions. The new goal is to carry out concurrent multi-disciplinary designs of the highly interactive systems in order to maximise aircraft performance, viewed in its broadest terms.

Achievement of this long term goal requires close collaboration between the major aeronautical disciplines: Aerodynamics; Structures; Load and Flutter; Propulsion; Guidance, Navigation and Control. Bearing this in mind, the flight control engineering discipline should utilise and elaborate controller analysis and design methodology suitable for multi-disciplinary considerations. Robust control methodology has this potential and is therefore the main focus of FM(AG08).

A major problem facing designers of Flight Control Systems (FCS) is uncertainty in characterising not only the vehicle itself, but also the environment in which it must operate. Gain scheduling is often necessary because of the variation of characteristics for which the control laws must guarantee stability and performance. The design of gain scheduling is costly for two reasons: the control laws must be designed at each design point, and a great deal of assessment is required to ensure adequate stability and performance at off-design points.

Recent advances in control theory research has given rise to a number of novel robust control techniques [6, 7] specifically developed for dealing with model uncertainties and parameter variations. These new techniques offer potential benefits to a control law designer for modern aircraft in the following ways:

- Multivariable systems can be handled in a concise methodical framework, thus removing the need for the sequential loop closure approach, and reducing the design effort required.

- Robust control laws which cover larger regions of the flight envelope around a design point can be derived more efficiently. This offers the potential for reducing the number of design points required, simplifying the gain schedule, and reducing the amount of assessment required at off-design points.

The main consequence of these benefits is that a FCS design based on robust control techniques yields a considerable reduction in the design effort required, and a potential reduction in the time-to-market and design costs. Subproject FM-AG08-3 (GARTEUR Robust Flight Control Design) aims at demonstrating these advantages to the European aeronautical industry.

1.2 Objectives of subproject FM-AG08-3

Robust control theory has been well assessed in the literature, where a great number of papers can be found dealing with the various aspects of robustness, parameter variations, modelling uncertainty, unmodelled dynamics, etc. At the same time a wide variety of algorithms implementing robust control techniques can be found in many technical reports and general control design and analysis software, such as Matlab/Simulink [16, 17] and $\mathrm{MATRIX}_\mathrm{X}$ [18].

However, robust control techniques are seldom used by European aircraft manufacturers for the design of FCS. There are three main reasons for this:

- Robust control theory is rather new and application to the aircraft control law design problem has not yet been demonstrated. The techniques and algorithms associated with robust control theory are clearly expressed but do not, in their current form, lend themselves to direct FCS application.
- There are a limited number of dedicated robust control design tools, while most manufacturers have an extensive suite of classical control design tools that they have developed over a period of several years. Robust control can also be achieved with classical methods, but indirectly making use of experience and analysis in an iteration loop.
- There is no specific bibliographic source available on robust control techniques. Consequently, a lot of time has to be spent in searching for appropriate references in a variety of widely distributed libraries, journals and general purpose data-bases.

Subproject FM-AG08-3 aims at removing these drawbacks and at demonstrating to European aircraft manufacturers that a significant improvement in the overall design process is possible by using robust control techniques. In a greater detail, the aim is:

- To identify and apply existing and new controller design methods to robust control problems that are representative of industrial needs [9].
- To introduce robust controller design and analysis methods into the control law design cycle, in order to cope more directly with uncertainty in the models used and with changes in flight conditions that occur.
- To identify tools which can be used in conjunction with multi-disciplinary design optimization to improve overall dynamic system performance.
- To develop robust controller design procedures that interface with industrial requirements.

To achieve these objectives, FM(AG08) has opted for the following approach. Two robust flight control benchmark problems have been defined, which will be solved by design teams from the European aeronautical industry, research establishments and universities. A wide variety of modern and classical design methods will be applied. The controllers that are designed in response to these problems will be compared and evaluated. A final conference will be held in Toulouse (France) in April, 1997, where the controllers and the results of the comparisons will be presented. However, it must be stated that the aim of these benchmark problems is not to produce an optimal control law, but to compare modern robust control theory with classical design methods applied to realistic problems. It is also intended that these benchmarks will raise the awareness and confidence of the European aeronautical industry in the use of robust control techniques.

The two benchmarks cover respectively an automatic landing control problem and a high angle of attack enhanced manual control problem. This document is the manual for the first problem, which will be referred to as the RCAM (Research Civil Aircraft Model) benchmark.

Participants are asked to design an autopilot for the final segments of an approach for a fictitious aircraft (RCAM). The control law must be robust with respect to variation of the speed, weight, variation of the horizontal and vertical position of the center of gravity, time delays, nonlinearities and engine failure. Disturbance decoupling must also be performed so that tracking of the glideslope and localiser paths must be within certain tolerances.

1.3 Contents of this document

The structure of the document is as follows:

- In chapter 2 a description of the RCAM model is given, in which analytical expressions for all the variables of interest, states, inputs and outputs of the system, are derived. A detailed description of the components of the model (aircraft, sensors, actuators and engines, wind model) is included.

- In chapter 3 the design problem is formulated, and the criteria and procedure adopted for evaluation of the proposed design are described.
- In chapter 4 the standard layout of the document that will contain the design results is given, with a description of the items to be addressed in each design document.
- In appendix A an installation procedure and user reference for the RCAM software model in Matlab/Simulink is given, together with examples.
- In appendix B an installation procedure and user reference for the software for writing the design document is given.
- In appendix C an installation procedure and user reference for the automated evaluation software is given, by which an auto-evaluation of the designed control law is possible.
- In appendix D the assessment software is described. This software is a more extensive suite of tests than the evaluation software.
- In appendix E the questionnaire used by the evaluation team is given.
- In appendix F a description of the Dymola model is given and how to automatically generate simulation software.

2 Description of the RCAM model

The purpose of this chapter is to discuss the RCAM model in a general setting, such that used nomenclature and terminology can be introduced, and some of the philosophy behind the structure and numerical calculations in the software can be highlighted. The chapter is set up to have some tutorial value, but is by no means complete in that sense. It is recommended to consult a standard reference such as [8] or [4] for more information on the derivation of equations of motion, etc.

2.1 Block diagram of the system

A six degree of freedom nonlinear model of the Research Civil Aircraft Model (RCAM) including nonlinearities of actuators (thresholds) and a model of disturbances has been proposed by Aérospatiale. A block diagram of the proposed model is given in figure 2.1. Each box in this block diagram will be covered in more detail in following text. In section 2.3 an analytical description of the aircraft dynamics is given. In sections 2.4 and 2.5 the sensor and actuator dynamics are detailed. In section 2.6 the analytical models of wind disturbances are presented.

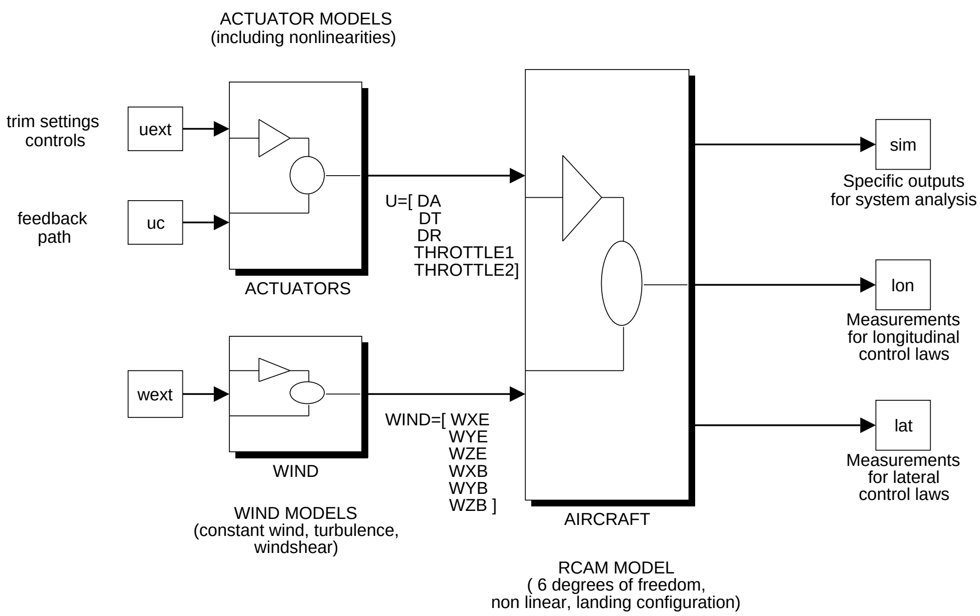

Fig.2.1 Block diagram of the system given as a Simulink representation [17].

2.2 Nomenclature: inputs, states, outputs, parameters

As far as applicable, nomenclature is used as defined in the Communication Handbook [2]. The following tables summarise this nomenclature, as it is used both for the formulation of the algorithms and the naming of variables in the software. Additional information can be found in Appendix A of this document.

The inputs to the model are given in table 2.1. In this table, F_E denotes the earth-fixed

Symbol	Alphanumeric			Name	Unit
Control inputs					
δ_A	DA	u(1)	=	aileron deflection	*rad*
δ_T	DT	u(2)	=	tailplane deflection	*rad*
δ_R	DR	u(3)	=	rudder deflection	*rad*
δ_{TH_1}	THROTTLE1	u(4)	=	throttle position of engine 1	*rad*
δ_{TH_2}	THROTTLE2	u(5)	=	throttle position of engine 2	*rad*
Wind inputs					
W_{XE}	WXE	u(6)	=	wind velocity in the x-axis of F_E	m/s
W_{YE}	WYE	u(7)	=	wind velocity in the y-axis of F_E	m/s
W_{ZE}	WZE	u(8)	=	wind velocity in the z-axis of F_E	m/s
W_{XB}	WXB	u(9)	=	wind velocity in the x-axis of F_B	m/s
W_{YB}	WYB	u(10)	=	wind velocity in the y-axis of F_B	m/s
W_{ZB}	WZB	u(11)	=	wind velocity in the z-axis of F_B	m/s

Table 2.1 Model inputs definitions

reference frame, which is defined as follows.

> The origin O_E is located on the runway longitudinal axis at the threshold. x_E is positive pointing towards the north, and we assume that the runway is also directed towards the north (runway 00), hence x_E is positive along the runway in the landing direction. Furthermore, z_E is positive downward, and y_E is in the appropriate direction for a right handed axis system (positive east).

F_B stands for the body-fixed reference frame, which is defined as follows.

> The origin O_B is at the vehicle centre of gravity. x_B is positive forward, z_B is positive downward and y_B is positive to the right (starboard side).

The three earth-fixed wind inputs, u(6)–u(8), are intended to be used for constant wind velocity components eg. headwinds, whereas the body-fixed wind inputs are intended to be used for gusts.

The states used internally by the software are expressed in SI units and are defined in table 2.2. In this table, 'CoG' denotes 'Centre of Gravity'.

Symbol	Alphanumeric			Name	Unit
p	P	x(1)	=	roll rate (in F_B)	rad/s
q	Q	x(2)	=	pitch rate (in F_B)	rad/s
r	R	x(3)	=	yaw rate (in F_B)	rad/s
ϕ	PHI	x(4)	=	roll angle (Euler angle)	rad
θ	THETA	x(5)	=	pitch angle (Euler angle)	rad
ψ	PSI	x(6)	=	heading angle (Euler angle)	rad
u_B	UB	x(7)	=	x component of inertial velocity in F_B	m/s
v_B	VB	x(8)	=	y component of inertial velocity in F_B	m/s
w_B	WB	x(9)	=	z component of inertial velocity in F_B	m/s
x	X	x(10)	=	x position of aircraft CoG in F_E	m
y	Y	x(11)	=	y position of aircraft CoG in F_E	m
z	Z	x(12)	=	z position of aircraft CoG in F_E	m

Table 2.2 States definitions

The outputs from the model are given in SI units and are shown in table 2.3. In this table, F_V denotes the vehicle-carried vertical frame, which is defined as follows.

> The vehicle-carried vertical frame is parallel to the earth-fixed reference frame but moves with the vehicle. The origin O_V is located at the vehicle's centre of gravity. x_V is positive pointing towards the north, z_V is positive downward, and y_V is positive towards the east.

Only the model outputs labeled as 'measured' can be assumed to be available as inputs to the controller that is to be designed. The 'simulation' outputs are only intended to be used for evaluation and should not be used for the controller. Note that there is some redundancy in the measured signals, e.g. χ can be determined from u_V and v_V: depending on the control strategy the most convenient signals may be used.

Usually, it is possible to define geometric aircraft parameters within the body-fixed reference frame. However, in the case of RCAM this is not allowed, as the CoG is not a geometrically fixed point. For this reason, a measurement reference frame F_M is defined.

> The measurement reference frame is geometrically fixed to the aircraft. The origin O_M is located at the leading edge of the mean aerodynamic chord, which is denoted as $\bar{c}$. x_M is positive pointing backwards, y_M is positive pointing to the right (starboard), and z_M is positive pointing up.

Symbol	Alphanumeric			Name	Unit
Measured					
q	Q	y(1)	=	pitch rate (in F_B) = x(2)	rad/s
n_x	NX	y(2)	=	horizontal load factor (in F_B) = $\frac{F_x}{mg}$	-
n_z	NZ	y(3)	=	vertical load factor (in F_B) = $\frac{F_z}{mg}$	-
w_V	WV	y(4)	=	z component of inertial velocity in F_V	m/s
z	Z	y(5)	=	z position of aircraft CoG in F_E = x(12)	m
V_A	VA	y(6)	=	air speed	m/s
V	V	y(7)	=	total inertial velocity	m/s
β	BETA	y(8)	=	angle of sideslip	rad
p	P	y(9)	=	roll rate (in F_B) = x(1)	rad/s
r	R	y(10)	=	yaw rate (in F_B) = x(3)	rad/s
ϕ	PHI	y(11)	=	roll angle (Euler angle) = x(4)	rad
u_V	UV	y(12)	=	x component of inertial velocity in F_V	m/s
v_V	VV	y(13)	=	y component of inertial velocity in F_V	m/s
y	Y	y(14)	=	y position of aircraft CoG in F_E = x(11)	m
χ	CHI	y(15)	=	inertial track angle	rad
Simulation					
ψ	PSI	y(16)	=	heading angle (Euler angle) = x(6)	rad
θ	THETA	y(17)	=	pitch angle (Euler angle) = x(5)	rad
α	ALPHA	y(18)	=	angle of attack	rad
γ	GAMMA	y(19)	=	inertial flight path angle	rad
x	X	y(20)	=	x position of aircraft CoG in F_E = x(10)	m
n_y	NY	y(21)	=	lateral load factor (in F_B)= $\frac{F_y}{mg}$	-

Table 2.3 Model outputs definitions

It is assumed that the aerodynamic centre of the wing-body configuration (AC_{wb}) is also geometrically fixed: its co-ordinates in F_M are $(0.12\bar{c}\,,\,0\,,\,0)$.
With these definitions, it is now possible to specify the parameters used in RCAM: they are given in table 2.4.

Finally, RCAM provides the possibility to study the effect of the parameter changes defined in table 2.5.

Symbol	Alphanumeric		Name	Default	Unit
Mass Parameters					
m	MASS	=	aircraft total mass	120 000	kg
Aerodynamic Parameters					
$\bar{c}$	CBAR	=	mean aerodynamic chord	6.6	m
l_t	LTAIL	=	distance between AC of the wing-body (AC_{wb}), and AC of the tail (AC_t)	24.8	m
S	S	=	wing planform area	260.0	m^2
S_t	STAIL	=	tail planform area	64.0	m^2
X_{cg}	XCG	=	x position of the CoG in F_M	0.23 $\bar{c}$	m
Y_{cg}	YCG	=	y position of the CoG in F_M	0	m
Z_{cg}	ZCG	=	z position of the CoG in F_M	0	m
Engine Parameters					
X_{APT1}	XAPT1	=	x position of application point of thrust of engine 1 in F_M	0.0	m
Y_{APT1}	YAPT1	=	y position of application point of thrust of engine 1 in F_M	−7.94	m
Z_{APT1}	ZAPT1	=	z position of application point of thrust of engine 1 in F_M	−1.9	m
X_{APT2}	XAPT2	=	x position of application point of thrust of engine 2 in F_M	0.0	m
Y_{APT2}	YAPT2	=	y position of application point of thrust of engine 2 in F_M	7.94	m
Z_{APT2}	ZAPT2	=	z position of application point of thrust of engine 2 in F_M	−1.9	m

Table 2.4 Parameters definitions

Parameters		Bounds					Nominal
m	MASS	100 000 kg	<	m	<	150 000 kg	120 000 kg
X_{cg}	XCG	0.15 $\bar{c}$	<	X_{cg}	<	0.31 $\bar{c}$	0.23 $\bar{c}$
Y_{cg}	YCG	−0.03 $\bar{c}$	<	Y_{cg}	<	0.03 $\bar{c}$	0.0 $\bar{c}$
Z_{cg}	ZCG	0.00 $\bar{c}$	<	Z_{cg}	<	0.21 $\bar{c}$	0.10 $\bar{c}$

Table 2.5 Possible parameter choices in RCAM, see also section 3.2.3.

2.3 Aircraft dynamics model

This section describes the RCAM dynamics model corresponding to the `AIRCRAFT` block in figure 2.1. The dynamic objects are depicted in figure 2.2.

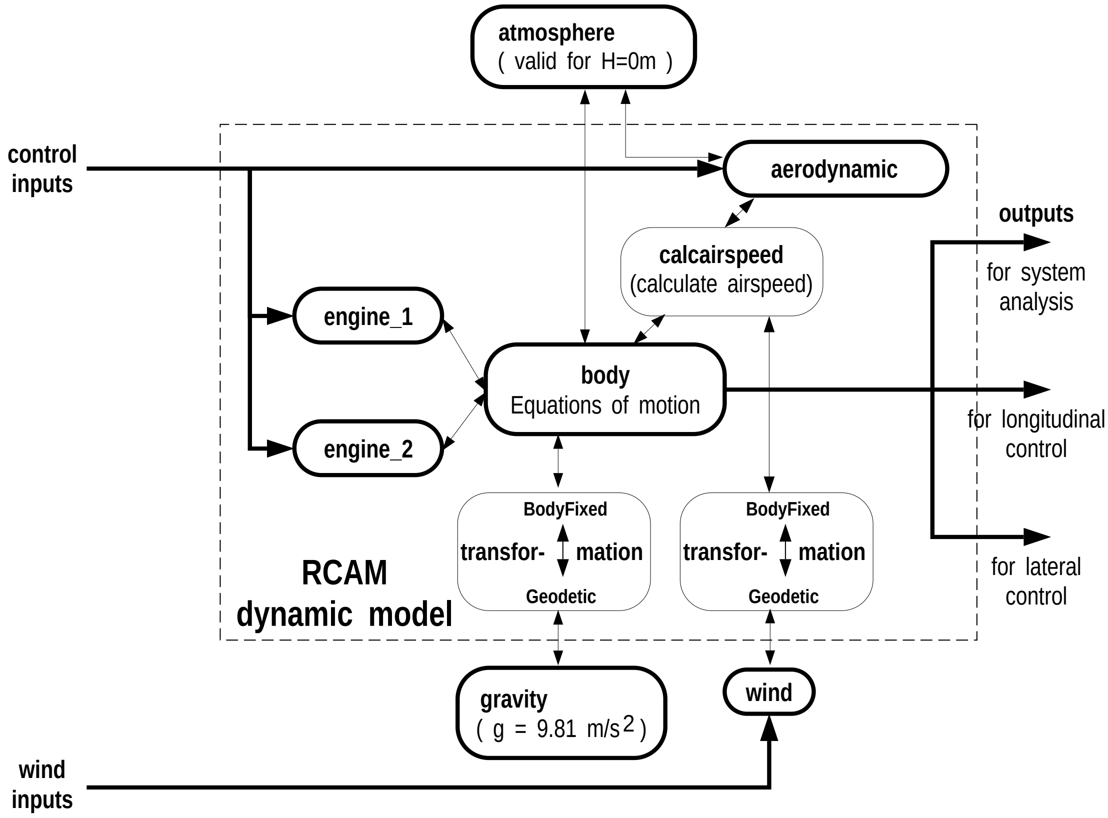

Fig.2.2 Dynamic objects of RCAM aircraft model inside the `AIRCRAFT` *block of figure 2.1. Connection arrows between objects characterise physical interactions*

These objects are:

- **body** describes the body differential equations of motion (see subsection 2.3.1);
- two **transformation** objects describe the co-ordinate transformation between the body-fixed co-ordinates of the body object and the geodetic co-ordinates of the gravity object, and between the body-fixed co-ordinates of body and the geodetic co-ordinates of wind, respectively (see subsection 2.3.2);
- **calcairspeed** describes the relationship between the inertial movement, the wind, and the movement relative to the air (see subsection 2.3.3);

- **engine_1** and **engine_2** describe the relevant engine behaviour (see subsection 2.3.5);
- **atmosphere** describes the atmosphere model (see subsection 2.3.6);
- **aerodynamic** describes the aerodynamic forces and moments (see subsection 2.3.4);
- **gravity** describes the gravitational influence (see subsection 2.3.7).

2.3.1 Body equations of motion

The following two subsections give a summary of the equations of motion for a rigid body with 6 degrees of freedom and other relevant equations. For a more detailed derivation and explanation of these equations a reference such as [8] should be consulted.

2.3.1.1 Translational motion

The equations for the translational movement in body-fixed co-ordinates are derived from the force vector equation,

$$F = m \left(a_B + \omega \times V_B \right) \tag{2.1}$$

where the vector F is the sum of forces due to the engines, the aerodynamics and gravity, m is the mass of the aircraft, V_B is the inertial velocity vector and ω is the rotational velocities expressed in body-fixed co-ordinates. The acceleration (in body-fixed system) is the time derivative of velocity:

$$a_B = \frac{d\,V_B}{dt} = \frac{d}{dt} \begin{bmatrix} u_B \\ v_B \\ w_B \end{bmatrix} \tag{2.2}$$

and the velocity is the time derivative of the position vector expressed in the vehicle-carried vertical frame:

$$V_V = \frac{d\,X_V}{dt} = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.3}$$

Additionally, some aircraft specific quantities are defined as follows:
The height h, which is the negative z-co-ordinate in the vehicle carried system

$$h = -z \tag{2.4}$$

The inertial flight path angle, γ, is given as a function of the speed components in the vehicle-carried vertical reference frame

$$\tan\gamma = \frac{-w_V}{\sqrt{{u_V}^2 + {v_V}^2}} \tag{2.5}$$

The track angle, χ, is also defined as a function of the the speed components in the vehicle-carried vertical reference frame

$$\tan\chi = \frac{v_V}{u_V} \tag{2.6}$$

2.3.1.2 Rotational motion

The equations of motion for the rotational movement of a rigid body in the body-fixed axis system are derived from the moment vector equation,

$$M = I\,\dot{\omega} \;+\; \omega \;\times\; I\,\omega \tag{2.7}$$

where M is the sum of moments about the centre of gravity due to the engines and the aerodynamics, ω is the inertial rotational velocity, and $\dot{\omega}$ is the inertial rotational acceleration in the body-fixed axis system. Using the standard notation [8] we get:

$$\dot{\omega} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.8}$$

Again using standard notation [8], the relation between the rotational velocities and the Euler angles is;

$$\frac{d\Phi}{dt} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.9}$$

For a normal aircraft*, the inertia tensor I defined in the body-axis frame is;

$$I \;=\; \begin{bmatrix} I_x & 0 & I_{xz} \\ 0 & I_y & 0 \\ I_{xz} & 0 & I_z \end{bmatrix} = m \begin{bmatrix} 40.07 & 0 & -2.0923 \\ 0 & 64 & 0 \\ -2.0923 & 0 & 99.92 \end{bmatrix} \tag{2.10}$$

where all numbers are expressed in square metres, m^2.

2.3.2 Co-ordinate transformation (Body-Fixed $\Leftrightarrow$ Vehicle-Carried)

The rotations between the body-fixed and the vehicle-carried co-ordinate system are depicted in figure 2.3.
To describe the angular orientation of the aircraft, a transformation using the three Euler angles ϕ, θ, and ψ is necessary. This transformation is achieved by initially rotating the vehicle-carried vertical system about the z_V-axis by the heading angle ψ. Then, the result is rotated about the k_2-axis by the pitch angle θ, and finally the body-fixed reference frame

*Normal aircraft are assumed symmetric about the OXZ body axis plane.

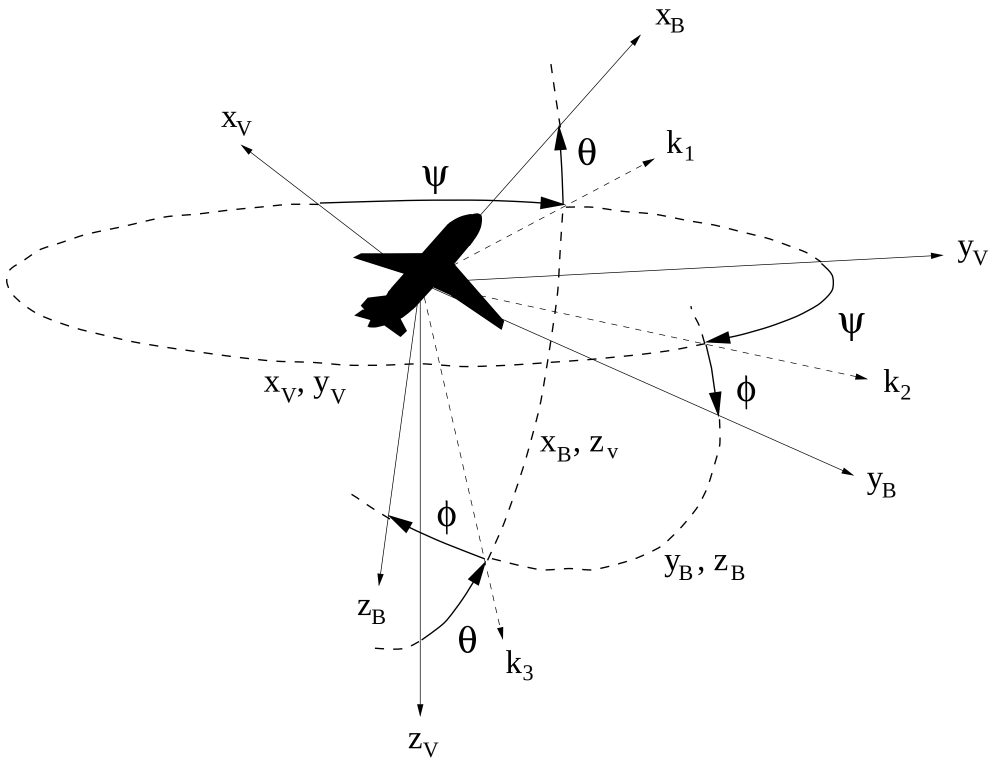

Fig.2.3 Co-ordinate transformation body-fixed ⇔ vehicle-carried

is obtained by rotating the result of that by the roll angle ϕ about the x_B-axis.

The transformation matrix from the vehicle-carried vertical axis system to the body-fixed axis system is given as follows:

$$R_{BV} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{2.11}$$

For example, the transformation of velocities from the vehicle-carried vertical frame F_V to the body-fixed reference frame F_B is given by:

$$V_B = R_{BV} V_V \tag{2.12}$$

with

$$V_B = \begin{bmatrix} u_B \\ v_B \\ w_B \end{bmatrix} \quad \text{and} \quad V_V = \begin{bmatrix} u_V \\ v_V \\ w_V \end{bmatrix} \tag{2.13}$$

Similarly, the accelerations, rotational velocities, positions, forces and moments can be transformed between the co-ordinate systems.

2.3.3 Calculation of airspeed

The vector airspeed, V_a is the difference between the inertial velocity of the aircraft, V_B, and the wind velocities, W_B and W_E (see table 2.1). Expressed in the body-fixed co-ordinate system this is calculated as:

$$V_a = V_B - W_B - R_{BV} W_E \tag{2.14}$$

Hence, with

$$V_a = \begin{bmatrix} u_a \\ v_a \\ w_a \end{bmatrix} \tag{2.15}$$

the airspeed V_A is given as:

$$V_A = \sqrt{({u_a}^2 + {v_a}^2 + {w_a}^2)} \tag{2.16}$$

Next, the angle of attack, α, and the angle of sideslip, β, are defined as:

$$\tan\alpha = \frac{w_a}{u_a} \tag{2.17}$$

$$\sin\beta = \frac{v_a}{V_A} \tag{2.18}$$

The derivatives of α and β with respect to time are:

$$\dot{\alpha} = \frac{a_{a_z}\, u_a \; - \; a_{a_x}\, w_a}{{u_a}^2 + {w_a}^2} \tag{2.19}$$

$$\dot{\beta} = \frac{a_{a_y}\, (\, {u_a}^2 \; + \; {w_a}^2 \,) \; - v_a\, (\, a_{a_x}\, u_a \; + \; a_{a_z}\, w_a \,)}{{V_A}^2\, \sqrt{{u_a}^2 + {w_a}^2}} \tag{2.20}$$

where a_{a_x}, a_{a_y}, and a_{a_z} are the x, y, and z-time derivatives of the airspeed in body-fixed co-ordinates. (e.g. $a_{a_x} = \frac{du_a}{dt}$).

2.3.4 Aerodynamic equations

The equations defining aerodynamic forces and moments are determined by means of aerodynamic coefficients. Depending on the method of modelling these coefficients may be defined in different reference frames; e.g. F_W, F_S, or F_B. The reference frame for aerodynamic forces and moments that is used in RCAM is the stability axis frame F_S.

2.3.4.1 Aerodynamic forces

The aerodynamic forces are determined by means of aerodynamic coefficients for drag, sideforce and lift (C_D, C_Y, C_L), which are given as functions of the angle of attack, α, the sideslip angle, β, and the control surface deflections. The data is given for landing configuration and is valid for low velocities with no Mach number variations.

The aerodynamic lift coefficient, C_L, is defined as (see figure 2.4);

$$C_L = C_{L_{wb}} + C_{L_t} \tag{2.21}$$

$C_{L_{wb}}$ is the lift coefficient of the wing and body. It acts on the aerodynamic centre of the wing and body and is parallel with the z_S axis. It is only a function of angle of attack α, and for $\alpha < 19$ degrees it is given by the following equations:

$$C_{L_{wb}} = \begin{cases} 5.5\ (\alpha - \alpha_0) & \alpha \leq 14.5\frac{\pi}{180} rad \\ -768.5\alpha^3 + 609.2\alpha^2 - 155.2\alpha + 15.2 & \alpha > 14.5\frac{\pi}{180} rad \end{cases} \tag{2.22}$$

Here, α_0 is the angle of attack at which the wing/body lift is zero:

$$\alpha_0 = -11.5\frac{\pi}{180} \tag{2.23}$$

The maximum lift coefficient is obtained at an angle of attack of $\alpha = 18$ degrees. Neglecting the effect of the tailplane, this is calculated from equation 2.22 as:

$$C_{L_{max}} = C_{L_{wb}}(\alpha = 18\frac{\pi}{180}) = 2.75 \tag{2.24}$$

The lift coefficient of the tailplane, C_{L_t} acts on the aerodynamic centre of the tailplane and is also parallel with the z_S axis. It is given as:

$$C_{L_t} = \frac{S_t}{S}\ 3.1\ \alpha_t \tag{2.25}$$

where α_t denotes the angle of attack of the tailplane and is calculated from the following equations:

$$\alpha_t = \alpha - \epsilon + \delta_T + 1.3\ \frac{q\ l_t}{V_A} \tag{2.26}$$

$$\epsilon = \frac{d\epsilon}{d\alpha}\ (\alpha - \alpha_0) \tag{2.27}$$

$$\frac{d\epsilon}{d\alpha} = 0.25 \tag{2.28}$$

Here ϵ is the downwash angle, δ_T is the tailplane deflection, q is the aircraft pitch rate, and l_t is the longitudinal distance between the aerodynamic centre of the tailplane and the aerodynamic centre of the wing and body. (See figure 2.4).

The aerodynamic drag coefficient, C_D, is a function of the angle of attack α; drag of the tailplane is neglected and it is assumed that C_D acts on the aerodynamic centre of wing and body:

$$C_D = 0.13 + 0.07 \cdot (5.5\alpha + 0.654)^2 \tag{2.29}$$

The aerodynamic sideforce coefficient, C_Y, is also assumed to act on the aerodynamic centre of wing and body and is given by the following equation:

$$C_Y = -1.6\beta + 0.24\delta_R \tag{2.30}$$

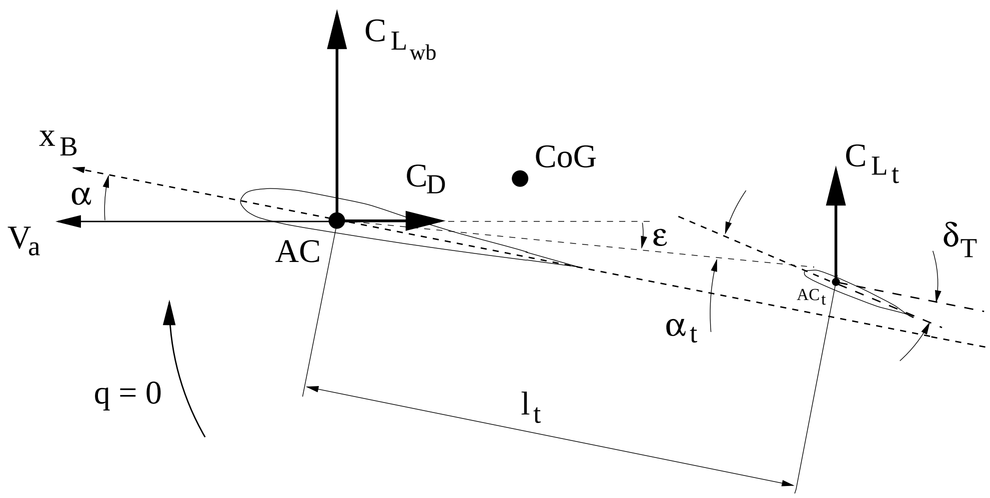

Fig.2.4 Illustration of aerodynamic forces

where β is the angle of sideslip and δ_R is the rudder deflection.

These non-dimensional coefficients can now be converted to dimensional forces using the following relationships:

- Aerodynamic force along x_S

$$X = -D = -C_D \, \frac{1}{2} \, \rho \, {V_A}^2 \, S \tag{2.31}$$

- Aerodynamic force along y_S

$$Y = C_Y \, \frac{1}{2} \, \rho \, {V_A}^2 \, S \tag{2.32}$$

- Aerodynamic force along z_S

$$Z = -L = -C_L \, \frac{1}{2} \, \rho \, {V_A}^2 \, S \tag{2.33}$$

To calculate the translational motion of the aircraft using equation 2.1, these forces need to be resolved into body axis force components. The resolution from stability axes forces, (D, Y, L), into the body-axes forces, (F_{xA}, F_{yA}, F_{zA}), is given by the following expressions:

$$F_{xA} = L \sin\alpha - D \cos\alpha \tag{2.34}$$
$$F_{yA} = Y \tag{2.35}$$
$$F_{zA} = -L \cos\alpha - D \sin\alpha \tag{2.36}$$

2.3.4.2 Aerodynamic moments

The moments due to the aircraft aerodynamics are determined by means of the moment coefficients, (C_l, C_m, C_n), which are assumed to act about the aerodynamic centre of the wing and body and are given by the following equation:

$$\begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} = \begin{bmatrix} -1.4\ \beta \\ -0.59 - 3.1\frac{S_t l_t}{S\bar{c}}(\alpha - \epsilon) \\ (1 - \alpha\frac{180}{15\pi})\beta \end{bmatrix} + \begin{bmatrix} -11 & 0 & 5 \\ 0 & -4.03\frac{S_t\ l_t^2}{S\ \bar{c}^2} & 0 \\ 1.7 & 0 & -11.5 \end{bmatrix} \frac{\bar{c}}{V_A} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} -0.6 & 0 & 0.22 \\ 0 & -3.1\frac{S_t\ l_t}{S\ \bar{c}} & 0 \\ 0 & 0 & -0.63 \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_T \\ \delta_R \end{bmatrix} \tag{2.37}$$

where
p, q,and r are the rotational rates in body axes,
δ_A is the aileron deflection,
δ_T is the tailplane deflection,
δ_R is the rudder deflection.

The moment coefficients about the centre of gravity are calculated from these aerodynamic centre based coefficients using the following equation:

$$\begin{aligned} \begin{bmatrix} C_{l_{CG}} \\ C_{m_{CG}} \\ C_{n_{CG}} \end{bmatrix} &= \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} + \frac{1}{\bar{c}} \begin{bmatrix} X_{cg} - 0.12\bar{c} \\ -Y_{cg} \\ Z_{cg} \end{bmatrix} \times \left(R_{BS} \cdot \begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix} \right) \\ &= \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} + \frac{1}{\bar{c}} \begin{bmatrix} 0 & C_Z & -C_Y \\ -C_Z & 0 & C_X \\ C_Y & -C_X & 0 \end{bmatrix} \begin{bmatrix} X_{cg} - 0.12\bar{c} \\ -Y_{cg} \\ Z_{cg} \end{bmatrix} \end{aligned} \tag{2.38}$$

with

$$R_{BS} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \qquad \begin{aligned} C_X &= -C_D\cos\alpha + C_L\sin\alpha \\ C_Z &= -C_L\cos\alpha - C_D\sin\alpha \end{aligned} \tag{2.39}$$

The following expressions are used to convert these non-dimensional moments coefficients into dimensional moments:

- Rolling moment in body axes

$$L_A = C_{l_{CG}}\ \frac{1}{2}\ \rho\ {V_A}^2\ S\ \bar{c} \tag{2.40}$$

- Pitching moment in body axes

$$M_A = C_{m_{CG}}\ \frac{1}{2}\ \rho\ {V_A}^2\ S\ \bar{c} \tag{2.41}$$

- Yawing moment in body axes

$$N_A = C_{n_{CG}}\ \frac{1}{2}\ \rho\ {V_A}^2\ S\ \bar{c} \tag{2.42}$$

These moments, in combination with the moments due to thrust are then used to calculate the rotational motion of the aircraft from equation 2.7.

2.3.5 RCAM engine thrust calculation

The RCAM is a twin engined aircraft model, and the thrust provided by each of the two engines is assumed to be aligned with the x-body axis. The thrust produced by a single engine is given by

$$F_i = \delta_{TH_i} mg, \qquad i = 1, 2 \tag{2.43}$$

with m at the nominal mass of 120,000 kg and δ_{TH_1} and δ_{TH_2} determined by the setting of the throttle handles. In equation 2.43, δ_{TH_i} should be expressed in radians: this has no physical meaning but appears to be convenient in the calculations. The allowed value of δ_{TH_i} lies between $0.5\,\frac{\pi}{180}$ and $10\,\frac{\pi}{180}$ radians. Note that the maximum thrust to weight ratio is about 0.35 (for both engines together). Hence, the engine thrust vector at the centre of gravity is given in F_B as:

$$F_p = \begin{bmatrix} F_1 + F_2 \\ 0 \\ 0 \end{bmatrix} \tag{2.44}$$

Due to the geometric location of the engines, see figure 2.5, the engine thrusts also contribute to the moments acting on the aircraft. These moments can be calculated about the centre of gravity as follows:

$$T_{Ei} = \begin{bmatrix} X_{cg} - X_{APTi} \\ Y_{APTi} - Y_{cg} \\ Z_{cg} - Z_{APTi} \end{bmatrix} \times \begin{bmatrix} F_i \\ 0 \\ 0 \end{bmatrix} \qquad (i = 1, 2) \tag{2.45}$$

where X_{cg}, Y_{cg}, Z_{cg}, X_{APTi}, Y_{APTi} and Z_{APTi} are defined in table 2.4.

2.3.6 Atmosphere

The atmosphere is considered to be constant irrespective of height and position. We use the following standard values for sea level:

$$\rho = 1.225 \frac{kg}{m^3} \tag{2.46}$$

$$P = 101325.0 \frac{N}{m^2} \tag{2.47}$$

$$T = 288.15K \tag{2.48}$$

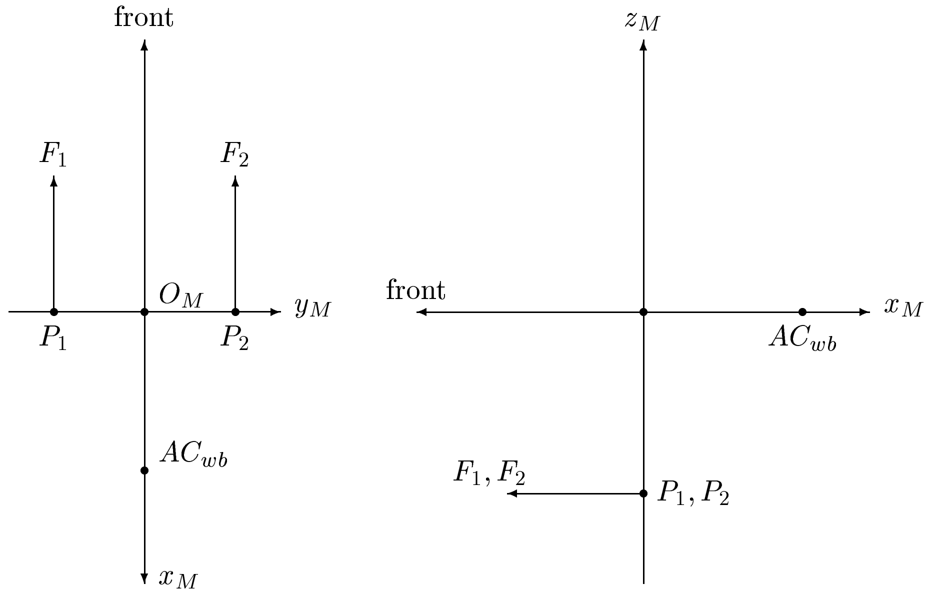

Fig.2.5 Application points of thrusts.
P_1 and P_2 are the points where the thrust is applied.

where
ρ is the density of air,
P is the static air pressure,
T is the absolute temperature.

2.3.7 Gravity model

Due to the restricted altitude range to be used with this model, gravity is not considered to be a function of altitude. Hence, gravity is assumed to have a constant value of:

$$g = 9.81\ m/s^2 \tag{2.49}$$

2.4 Sensor models

Models are not provided for the characteristics of the sensors: they are all assumed to be perfect.

2.5 Actuator models and engine dynamics

The Simulink diagram given in figure 2.5 shows the implementation of the actuator and engine models.
They all are assumed to have first order system dynamics with rate limits and saturations. The time constants of the first order system dynamics are:

- engine models: 1.5 s,

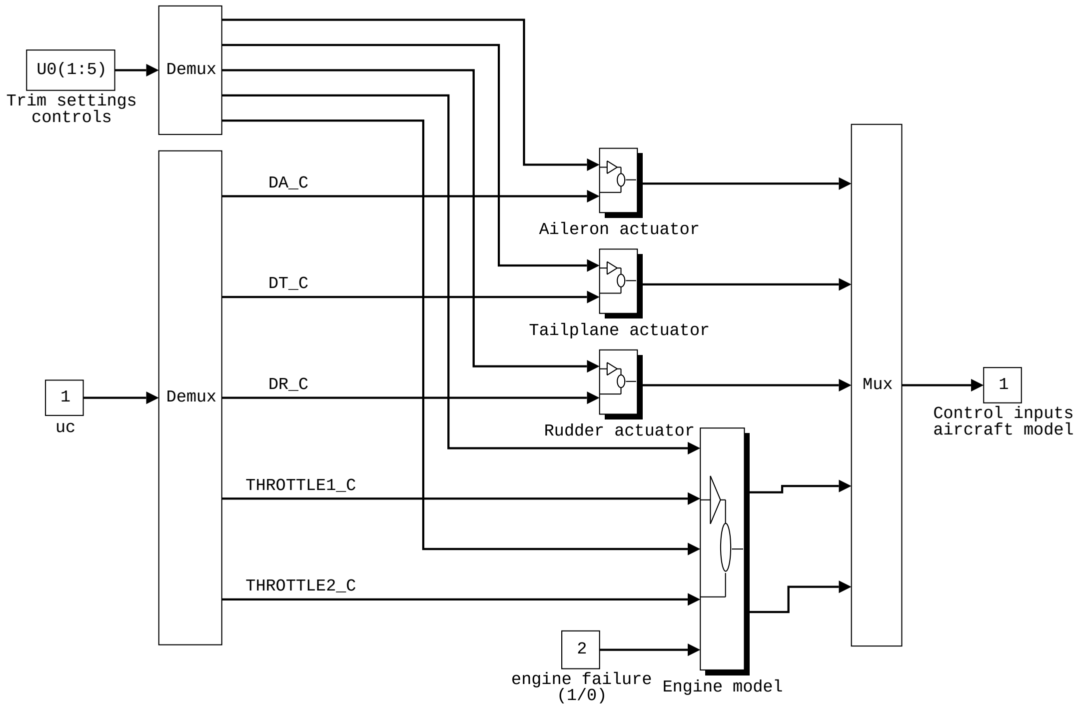

Fig.2.6 Actuator models

- ailerons and tailplane actuators: 0.15 s, and
- rudder actuator: 0.3 s.

Numerical values for rate limits and saturations are given as follows.

- Rate limits for throttle movement are:
  rising slew rate $= 1.6\frac{\pi}{180}$ rad/s, falling slew rate $= -1.6\frac{\pi}{180}$ rad/s,
- throttle limits (saturations) are: $0.5\frac{\pi}{180}$ $rad \leq \delta_{TH_i} \leq 10\frac{\pi}{180}$ rad.

In case of engine failure we can assume that the throttle setting for the failed engine reduces to $\delta_{TH_i} = 0.5\frac{\pi}{180}$ rad with first order system dynamics given by the transfer function $1/(1+3.3s)$.

- Rate limits for aileron deflection are: $-25\frac{\pi}{180} \leq \dot{\delta}_A \leq 25\frac{\pi}{180}$ rad/s;
  saturations of aileron deflection are: $-25\frac{\pi}{180} \leq \delta_A \leq 25\frac{\pi}{180}$ rad,
- rate limits for tailplane deflection are: $-15\frac{\pi}{180} \leq \dot{\delta}_T \leq 15\frac{\pi}{180}$ rad/s;
  saturations of tailplane deflection are: $-25\frac{\pi}{180} \leq \delta_T \leq 10\frac{\pi}{180}$ rad,
- rate limits for rudder deflection are: $-25\frac{\pi}{180} \leq \dot{\delta}_R \leq 25\frac{\pi}{180}$ rad/s;
  saturations of rudder deflection are: $-30\frac{\pi}{180} \leq \delta_R \leq 30\frac{\pi}{180}$ rad.

2.6 Atmospheric turbulence model

2.6.1 Turbulence spectra

Turbulence is a stochastic process that can be defined by velocity spectra. Commonly used velocity spectra for turbulence modelling are the Dryden spectra. For an aircraft flying at a speed V through a 'frozen' turbulence field with a spatial frequency of Ω *rad/m*, the circular frequency of the turbulence can be calculated as:

$$\omega = V \cdot \Omega \quad rad/s \tag{2.50}$$

With this, according to [31] the spectra can be described as follows:

$$\begin{aligned}
\Phi_{u_g}(\omega) &= \sigma_u^2 \frac{2L_u}{\pi V} \frac{1}{(1 + (L_u \frac{\omega}{V})^2)} \\
\Phi_{v_g}(\omega) &= \sigma_v^2 \frac{2L_v}{\pi V} \frac{1 + 12(L_v \frac{\omega}{V})^2}{(1 + 4(L_v \frac{\omega}{V})^2)^2} \\
\Phi_{w_g}(\omega) &= \sigma_w^2 \frac{2L_w}{\pi V} \frac{1 + 12(L_w \frac{\omega}{V})^2}{(1 + 4(L_w \frac{\omega}{V})^2)^2}
\end{aligned} \tag{2.51}$$

The turbulence scale lengths L_u, L_v, L_w and turbulence standard deviations σ_u, σ_v, σ_w are dependent on altitude and atmospheric conditions.

2.6.2 Turbulence standard deviation

According to [31], the turbulence standard deviations are defined in statistical terms. Common indications with probability of exceedance are:

light	10^{-2}
moderate	10^{-3}
severe	10^{-5}

For $3 < h < 300$ m the standard deviations for the given statistical categories are defined by:

$$\begin{aligned}
\sigma_u &= \sigma_v = \frac{\sigma_w}{(0.177 + 0.00274\ h)^{0.4}} \\
\sigma_w &= 0.8\ m/s \quad \text{light} \\
\sigma_w &= 1.6\ m/s \quad \text{moderate} \\
\sigma_w &= 2.3\ m/s \quad \text{severe}
\end{aligned} \tag{2.52}$$

For medium to high altitudes the standard deviations for the three components are equal to each other. The dependency from altitude for the given statistical categories is defined by:

- light

$$
\begin{aligned}
\sigma &= 1.55\ m/s & 600 \leq h \leq 2800\ m \\
\sigma &= 2.32 - 0.000274\ h & 2800 < h < 5100\ m \\
\sigma &= 0.92\ m/s & h \geq 2800\ m
\end{aligned} \tag{2.53}
$$

- moderate

$$
\begin{aligned}
\sigma &= 3.05\ m/s & 600 \leq h \leq 3400\ m \\
\sigma &= 3.84 - 0.000234\ h & h > 3400\ m
\end{aligned} \tag{2.54}
$$

- severe

$$
\begin{aligned}
\sigma &= 3.04 + 0.00244\ h & 600 < h < 1400\ m \\
\sigma &= 6.45\ m/s & 1400 \leq h \leq 5800\ m \\
\sigma &= 8.40 - 0.000336\ h & h > 5800\ m
\end{aligned} \tag{2.55}
$$

For the missing part from 300 to 600 m altitude linear interpolation is suggested:

$$
\begin{aligned}
\sigma &= 0.05 + 0.0025\ h & \text{light} \\
\sigma &= 0.15 + 0.00483\ h & \text{moderate} \\
\sigma &= 0.1 + 0.00733\ h & \text{severe}
\end{aligned} \tag{2.56}
$$

2.6.3 Turbulence scale length

The turbulence scale lengths L_u, L_v and L_w are defined in [31] as functions of altitude. For $3 < h < 300\ m$ it is given:

$$
\begin{aligned}
L_u &= 2\ L_v = \frac{h}{(0.177 + 0.00274\ h)^{1.2}} \\
L_w &= \frac{h}{2}
\end{aligned} \tag{2.57}
$$

For $h > 600\ m$ the scale lengths are defined by:

$$
L_u = 2\ L_v = 2\ L_w = 530\ m \tag{2.58}
$$

and for the missing part from 300 to 600 m altitude linear interpolation is suggested:

$$
L_u = 2\ L_v = 2\ L_w = 70 + 0.766\ h \tag{2.59}
$$

2.6.4 Turbulence simulation

To simulate turbulence, white noise is filtered through shaping filters. These filters can be derived from the Dryden spectra given in equation 2.51. As an example, the transfer function of the filter for simulating the gust velocity w_g will be considered.
Given white noise wn, the spectrum of w_g can be obtained as:

$$\Phi_{w_g} \;=\; |H_{w_g wn}(\omega)|^2 \; \Phi_{wn} \tag{2.60}$$

Where Φ_{wn} has unit power spectral density over a relatively wide bandwidth, and $H_{w_g wn}(\omega)$ is the frequency response function of the shaping filter. Therefore,

$$\sigma_w^2 \frac{2L_w}{\pi V} \frac{1+12(L_w \frac{\omega}{V})^2}{(1+4(L_w \frac{\omega}{V})^2)^2} \;=\; |H_{w_g wn}(\omega)|^2 \;=\; H_{w_g wn}(\omega)\; H_{w_g wn}(-\omega) \tag{2.61}$$

To obtain a stable and minimum phase filter, the frequency response function $H_{w_g wn}(\omega)$ is selected, resulting in the following transfer function:

$$H_{w_g wn}(s) \;=\; \sigma_w \sqrt{\frac{2L_w}{\pi V}} \frac{1+2\sqrt{3}\frac{L_w}{V}s}{(1+2\frac{L_w}{V}s)^2} \tag{2.62}$$

The transfer function for generating v_g is equivalent.
The transfer function for generating u_g can be found as:

$$H_{u_g wn}(s) \;=\; \sigma_u \sqrt{\frac{2L_u}{\pi V}} \frac{1}{1+\frac{L_u}{V}s} \tag{2.63}$$

It is important to note that for correct application of these filters the white noise inputs need to be independent.
With this procedure, the gust velocities u_g, v_g and w_g are defined in the stability reference frame. However, as an approximation the RCAM inputs W_{XB}, W_{YB} and W_{ZB} are used. For a more detailed discussion on turbulence modelling, the reader is referred to for example [4].

3 Design problem formulation and evaluation criteria

3.1 Motivation design and evaluation criteria

Within the aerospace industry there is a large amount of experience in the flight control system design area . For this reason, the main objective of the control problem stated here is not so much to obtain a satisfactory controller, but more specifically:

GC1: to exhibit approaches which might reduce the complexity of control laws and the overall control system design cycle.

Here GC denotes a general criterion.
Some of the main features addressed by modern control design techniques provide the possibility to take into account:

- the multivariable nature of the control problem
- the nonlinear behaviour of the plant
- the time-varying nature of the plant
- robustness to parameter changes and uncertainties
- simultaneous performance and robustness specifications.

From the consideration of these features it is expected that improvements could be made in areas such as:

- control system architecture development
- control law design cycle
- control design solution
- control system implementation

The RCAM design challenge consists of the synthesis of a control law capable of fulfilling an approach to landing under various external conditions eg. turbulence and windshear, while being robust to parameter changes. Furthermore,

GC2: the aircraft guidance must not degrade under engine failure.

Details on the design objectives are given in section 3.2.
For the uniform comparison of all design entries from the design challenge participants, a set of evaluation criteria is formulated in section 3.3. There are essential differences between the design specifications and the controller properties tested in the evaluation procedure. The evaluation procedure is not designed to guarantee fulfillment of all design

specifications, but to give a relative measure between different and dissimilar designs. However, we must emphasize that the design criteria and not the evaluation procedure should be the basis for the design. Also, the evaluation procedure does not cover all requirements.
To evaluate proper control system logic and to make the challenge more realistic, an evaluation trajectory has been designed to reflect typical phases during approach to landing. The evaluation criteria given in this section are based on sets of signals from which certain characteristics will be calculated.

GC3: All designs should be able to track the given trajectory within the specified bounds as defined in table 2.2.

Note that the choice of a trajectory as an evaluation criterion is independent of the control law and control design methodology.
An important subject considered in this chapter is the translation of design objectives into evaluation criteria: the evaluation criteria should be sufficiently representative for the considered design objectives, but will not be able to cover all aspects. It is asked that the benchmark problem participants consider the design objectives given in section 3.2 and for them to use their own methods to illustrate to what extent these are met by their controller design. For instance, we give robustness specifications in terms of real parameter variations, although they are often also considered in the frequency domain or in terms of gain and phase margins. The evaluation procedure is only aimed at obtaining an objective measure for comparison with other designs.

3.2 Design criteria

3.2.1 Introduction

The controller design problem for the RCAM model is characterised by a number of fundamental trade-offs between conflicting design specifications. For typical aircraft autopilot systems we recognise five classes of criteria:

performance criteria (PC): these reflect tracking error and disturbance rejection characteristics of certain signals;

robustness criteria (RC): these reflect the stability bounds with respect to parameter variations;

ride quality criteria (RQC): these reflect the desire to obtain sufficient passenger and pilot comfort in the form of bounds on certain maximum allowable accelerations and minimum damping levels;

safety criteria (SC): these reflect envelope safeguards;

control activity criteria (CAC): these are a measure of the power consumed by the coantrols and also give an indication of fatigue effects.

3.2.2 Performance criteria (PC)

The performance of the controlled system can be specified in terms of command response characteristics to normalised reference signals, tracking error and disturbance rejection features (see [11]). The step command response characteristics are defined in terms of rise time t_r, settling time t_s and overshoot M_p. Rise time is defined here as the time the unit step response $y(t)$ takes from $y = 0.10$ to $y = 0.90$, i.e., $t_r = t(y_{90\%}) - t(y_{10\%})$, see figure 3.1. Settling time is here defined as the time for $y(t)$ to achieve 99 percent of its final value. Finally, overshoot is defined as the relative peak of $y(t)$, i.e., $M_p = \frac{(y_{peak} - y(\infty))}{y(\infty)} \times 100\%$ (see [10]).

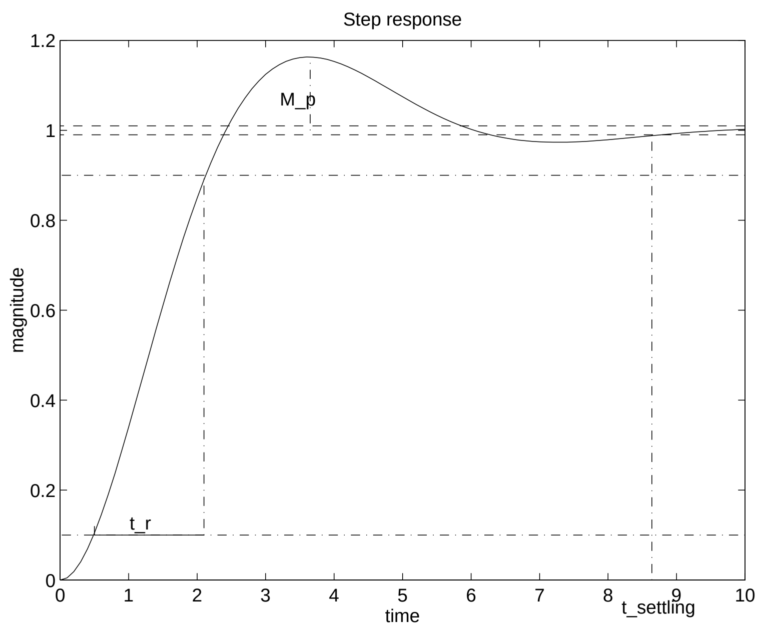

Fig.3.1 Unit step response. The rise time t_r is defined as the time from $y = 0.10$ to $y = 0.90$. The settling time is defined as the time for $y(t)$ to achieve 99% of its final value. The overshoot is defined as the relative peak of $y(t)$.

PC1 Lateral deviation

PC1.1 The controlled aircraft's lateral deviation, $e_{yb}(t)$, defined as the difference between the actual and commanded lateral aircraft position, $y(t) - y_c(t)$, should be reduced to 10 percent within 30 s.

PC1.2 There should be very little overshoot in the response to a unit step in lateral command signals at altitudes above 305 m (1000 ft), i.e., $M_p < 5\%$.

PC1.3 At lower altitudes M_p may increase to 30% in order to obtain higher tracking performance.

PC1.4 There should be no steady state error due to constant lateral wind disturbances.

PC1.5 In the final phase of flight (landing approach glide path) the lateral deviation from the desired flight path should not exceed that given in figure 3.2.

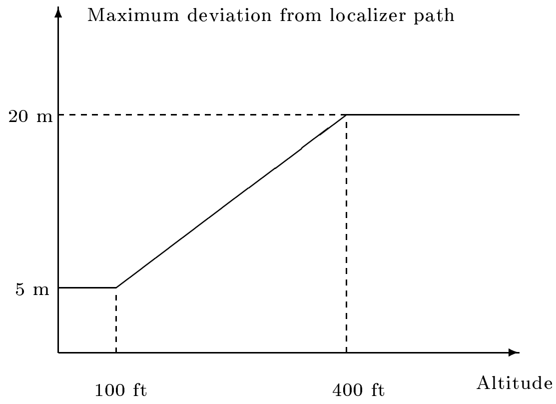

Fig.3.2 Maximum lateral deviation (PC1.5)

PC2 Altitude response

PC2.1 The controlled system should be able to track altitude commands, h_c, with rise time $t_r < 12\ s$,

PC2.2 and settling time $t_s < 45\ s$.

PC2.3 There should be very little overshoot in the response to unit steps in altitude commands at altitudes above 305 m (1000 ft), i.e., $M_p < 5\%$.

PC2.4 At lower altitudes M_p may increase to 30% in order to obtain higher tracking performance.

PC2.5 In the final phase of flight (landing approach glide path) the vertical deviation from the desired flight path should not exceed that given in figure 3.3.

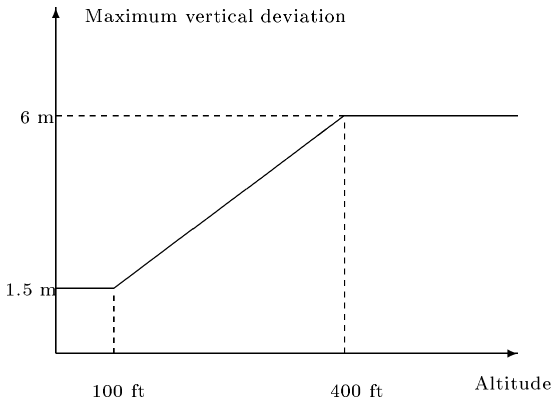

Fig.3.3 Maximum vertical deviation (PC2.5)

PC3 Heading angle response

PC3.1 The commanded heading angle, ψ_c, should be tracked by the actual heading angle, ψ, with a rise time $t_r < 10\ s$

PC3.2 and and settling time $t_s < 30\ s$.

PC3.3 There should be very little overshoot in the response to unit steps in heading commands at altitudes above 305 m (1000 ft), i.e., $M_p < 5\%$.

PC3.4 At lower altitudes M_p may increase to 30% in order to obtain higher tracking performance.

PC3.5 For unit RMS intensity lateral Dryden gust the RMS of the heading angle error in closed loop should be less than that in open loop.

PC4 Flight path angle response

PC4.1 The commanded flight path angle, γ_c, should be tracked by the actual flight path angle, γ, with a rise time $t_r < 5\ s$,

PC4.2 and settling time $t_s < 20$ *s*.

PC4.3 There should be very little overshoot in the response to unit steps in flight path angle commands at altitudes above 305 *m* (1000 *ft*), i.e., $M_p < 5\%$.

PC4.4 At lower altitudes M_p may increase to 30% in order to obtain higher tracking performance.

PC5 Roll angle response

PC5.1 In case of engine failure in still air, the roll angle, ϕ, should not exceed 10 *deg*;

PC5.2 its maximum steady state deviation should not exceed 5 *deg*.

PC5.3 During engine failure, sideslip angle β should be minimised;

PC5.4 the steady state roll angle that is needed to achieve this, should be reduced to zero with an overshoot of less than 50% when the failed engine is restarted (the failed engine's throttle setting steps back to that of the active engine).

PC5.5 Under moderate turbulence conditions (see section 2.6) ϕ should remain smaller than 5 *deg*.

PC6 Airspeed response

PC6.1 The controlled system's airspeed, V_A, should be able to track speed commands, V_{Ac}, with a rise time $t_r < 12$ *s*,

PC6.2 and settling time $t_s < 45$ *s*.

PC6.3 There should be very little overshoot in the step response to speed commands at altitudes above 305 *m* (1000 *ft*), i.e., $M_p < 5\%$.

PC6.4 At lower altitudes M_p may increase to 30% in order to obtain higher tracking performance.

PC6.5 In the presence of a wind step with an amplitude of 13 *m/s* (25 *kts*) there should be no deviation in the airspeed larger than 2.6 *m/s* (5 *kts*) for more than 15 *s*.

PC6.6 There should be no steady state error due to constant wind disturbances.

Note that a wind step will produce a corresponding initial step in airspeed, V_A, but no initial change of inertial velocity, V. However, as airspeed, V_A, is controlled back to its commanded value, the inertial velocity, V, will change by the value of the wind step.

PC7 Heading rate

PC7.1 In case of engine failure, the maximum heading rate, $\dot{\psi}$, should be less than 3 *deg/sec*.

PC8 Cross coupling between airspeed V_A and altitude h

PC8.1 For a step in commanded altitude h_c of 30 m, the peak value of the transient of the absolute error between V_A and commanded airspeed V_{Ac} should be smaller than 0.5 m/s (1 kt).

PC8.2 Conversely, for a step in commanded airspeed V_{Ac} of 13 m/s (25 kts), the peak value of the transient of the absolute error between h and h_c should be smaller than 10 m.

3.2.3 Robustness criteria (RC)

RC1 Centre of gravity variation Stability and sufficient performance should be maintained for horizontal centre of gravity variations between 15% and 31% $\bar{c}$,

RC2 and vertical centre of gravity variations between 0 and 21% $\bar{c}$ of the mean aerodynamic chord (see table 2.5; we will not consider variations in lateral direction).

RC3 Mass variations Stability and sufficient performance should be maintained for aircraft mass variations between 100000 to 150000 kg.

RC4 Time delay Stability and sufficient performance should be maintained for transport delays from 0 to 100 ms.

RC5 Speed variations Stability and sufficient performance should be maintained for speed variations from $1.23V_S$ to 90 m/s.

3.2.4 Ride quality criteria (RQC)

Ride quality criteria (RQC) should ensure sufficient passenger and pilot comfort. The following specifications are designed to obtain an acceptable level.

RQC1 Maximum vertical acceleration
Under normal conditions (no turbulence) the *vertical* acceleration at the centre of gravity should be minimised. During straight flight it should be less than $\pm$ 0.05 g[*], and during a 30-degree turn less than $\pm$ 0.2 g.

RQC2 Maximum lateral acceleration
Under normal conditions during manoeuvres (no turbulence) the *lateral* acceleration at the centre of gravity should be minimised. During straight flight it should be less than $\pm$ 0.02 g, and during a 30-degree turn less than $\pm$ 0.04 g.

[*] This value is used in industry during the design phase, in fact the vertical and lateral acceleration limits depend on frequency. They are even lower at 2 Hz.

RQC3 Damping

RQC3.1 Unless stated differently, there should be no overshoot in any step response of any controlled variable at altitudes above 305 m (1000 ft).

RQC3.2 Below that altitude overshoot may increase to 30% in order to obtain higher tracking performance.

3.2.5 Safety criteria (SC)

SC1 Airspeed The airspeed must always be larger than $1.05 \times V_{\text{stall}}$, where V_{stall} denotes the stall speed, i.e. the speed below which the aircraft is unable to maintain flight. This speed can be found from the following equilibrium relation:

$$mg = \frac{1}{2}\rho S V_{\text{stall}}^2 \, C_{L_{max}} \tag{3.1}$$

Substituting the relevant values from chapter 2, and assuming a mass of 120000 kg, we obtain $V_{\text{stall}} = 51.8$ m/s.

SC2 Angle of attack In section 2.3.4 it was given that the maximum lift coefficient is obtained at an angle of attack of 18 degrees. Hence, it can be concluded from the previous requirement that the stall speed corresponds to: $\alpha_{\text{stall}} = 18$ deg. A value of 12 deg is considered acceptable.

SC3 Roll angle The maximum roll angle ϕ should be limited to 30 deg.

SC4 Sideslip angle response At all times, sideslip angle β should be minimised. For unit RMS intensity lateral Dryden gust the RMS of the sideslip angle in closed loop should be less than that in open loop.

3.2.6 Control activity criteria (CAC)

CAC1 Actuator effort minimisation

CAC1.1 Under moderate turbulence conditions (see section 2.6), mean actuator rates for aileron,

CAC1.2 tailplane

CAC1.3 and rudder should be less than 33% of the maximum rates (see section 2.5).

CAC2 Engine effort minimisation
Under moderate turbulence conditions (see section 2.6), mean throttle rate should be less than 15% of the maximum rate (see section 2.5).

3.3 Evaluation procedure: RCAM mission and scenario

To be able to evaluate all kinds of different control design methods and resulting controllers it is necessary to find a uniform evaluation procedure, independent of the design method. A procedure to do this is to define a mission and a typical landing approach scenario (see [14, 5, 15]). This mission consists of manoeuvres that can be evaluated by means of nonlinear simulations. The performance of the control law depends on the mission phase, within which hard criteria or bounds on certain signals should be met and/or error signals must be minimised.

Note that not all of the given design criteria are checked by this evaluation procedure, especially only a few of the uncertain parameters test cases are checked. It is up to the designer to guarantee that all design criteria are satisfied.

The mission and scenario to be 'flown' by the RCAM model consists of a landing approach divided into the following segments, see figure 3.4.

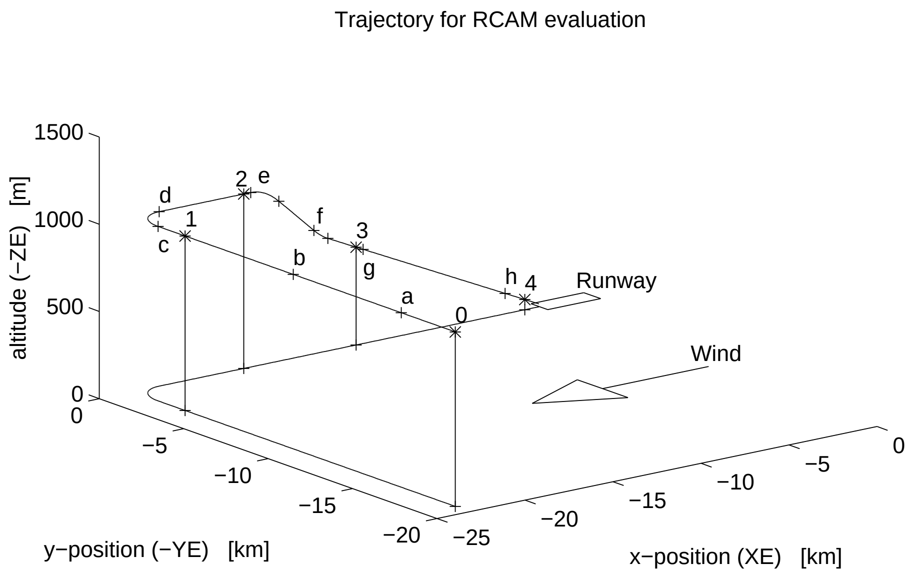

Fig.3.4 The landing approach for RCAM. The final leg is heading towards the North.

Segment I (0 to 1) Starting at an altitude of 1000 m and with a track angle of $\chi = -90$ deg (270^o west), level flight is to be maintained with a constant airspeed of 80 $m/s^\dagger$. During this segment, the lateral features of the autopilot will be investigated by simulating failure of the left engine (engine 1). This is indicated in figure 3.4: the failure occurs at point **a**, after which the engine is restarted at point **b**. The transient and steady state behaviour of the system will be analysed.

Segment II (1 to 2) This segment consists of a commanded co-ordinated turn from

†The airspeed during the landing phase depends on the aircraft mass, it is taken equal to 1.3 times V_{stall}: with a maximum landing weight of 150000 kg this results in ≈ 80 m/s.

points **c** to **d** with a heading rate of $\dot{\psi} = 3$ *deg/sec*. The objectives are to maintain a constant speed of 80 *m/s*, to keep the lateral acceleration close to zero, and to restrict the bank angle to $\phi = 30$ *deg* with consistent rudder/aileron deflections.

Segment III (2 to 3) For the descent phase a two-segment approach procedure will be used with $\gamma = -6$ *deg* and $\gamma = -3$ *deg*, which has been proposed for reasons of environmental noise reduction. This descent procedure is not standard but it is a better evaluation test than the standard ILS procedure, which has a constant glide slope angle of $\gamma = -3\,deg$. The starting altitude is $h = 1000$ *m*. After a short period of level flight, the flight path angle is set to $\gamma = -6$ *deg* at point **e**, and to $\gamma = -3$ *deg* at point **f**. The desired airspeed is 80 *m/s*.

Segment IV (3 to 4) The glide slope of $\gamma = -3$ *deg* is to be maintained during a wind shear between points **g** and **h**. The aircraft has to maintain safe flight and should not deviate too far from the desired glide path. The wind shear model used in the evaluation procedure is a two dimensional model derived from [13] (also see section 3.4.4 for more information). The desired airspeed is 80 *m/s*.

Throughout the evaluation procedure a Dryden turbulence field, of scale length $L = 305\,m$ and amplitude $\sigma = 0.08$ *m/s*, is assumed to be active. Note that the amplitude is only 5% of the amplitude for moderate conditions as defined in Chapter 2: this is done to prevent that the effect of turbulence on lateral and longitudinal accelerations overrules other effects that we are interested in. Superimposed on top of this turbulence is a 10 *m/s* constant wind with a fixed heading. This constant wind is active in full respect during Segments I and II until point **d**, and is slowly reduced to zero between points **d** of Segment II and **g** of Segment IV (at the start of the wind shear model). The wind has no vertical component and is directed along the negative earth-fixed x-axis, i.e., it is a cross-wind during Segment I and a headwind during Segment III.

To check robustness properties with respect to x-position of centre of gravity and to time delays, the entire approach will be flown in four different test cases. The horisontal location of the centre of gravity has three positions: most forward, nominal and a most aft. Furthermore, one flight will be executed with a nominal centre of gravity location and a time delay of 100 *ms*.

3.4 Spot-checking the design criteria by equivalent evaluation criteria

It should be noted that it is not possible to check all desired autopilot features by flying a single landing approach trajectory. Furthermore, the evaluation procedure should be relatively simple and straightforward: we want to be able to apply it to a great variety of controllers. Hence, the evaluation criteria should be independent of the type of controller used: they should consist of calculable indicators that enable us to obtain an objective comparison between completely different controllers.

For these evaluation criteria we will use the same classification as was given in the definition of the design criteria.

- performance
- robustness
- ride quality
- safety
- control activity

For each of these items and for each of the four trajectory segments a single number will be calculated. This number should not be considered to be the final assessment of total autopilot performance: it is merely an indicator for one or two important aspects. In most cases it is chosen such that a value of smaller than one is acceptable.
To further evaluate the dynamic behaviour of the autopilot, we will consider several plots of key variables during each of the segments. We will compare the shape of the actual trajectory with the demanded trajectory and provide bounds that should be respected for good performance. Similarly, we will plot the most important deviations from the desired trajectory.

3.4.1 Segment I

For segment I we will plot a plan view of the reference trajectory and the four test cases defined in section 3.3, and then superimpose the bounds given in figure 3.5. The points **a**

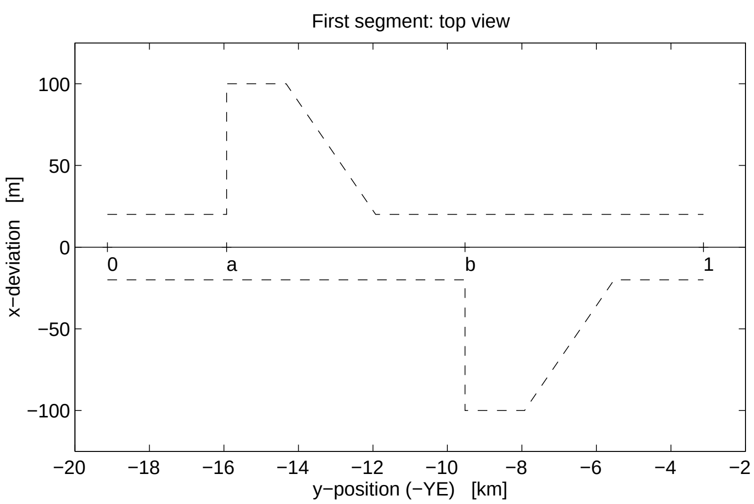

Fig.3.5 Segment I: the effect of engine failure with bounds

and **b** correspond to the beginning and end of the engine failure segment.

Performance.

The bound of 20 m for the lateral deviation is given to account for the effect of turbulence. During engine failure, we allow a maximum lateral deviation of 100 m that should be quickly reduced to less than 20 m at the end of the segment (when the aircraft should be stabilised again). With e_{yb} denoting the lateral deviation in body co-ordinates for the trajectory with nominal centre of gravity and time delay we will use

$$\frac{1}{2}\max_t\left(\frac{|e_{yb}(t)|}{100}+\frac{|e_{yb}(t_1)|}{20}\right) \tag{3.2}$$

as a measure that should be smaller than one for sufficient performance. Note that during the entire segment the maximum deviation of 100 m should not be exceeded, and that at the end of the segment (t_1 corresponds with point **1**) the maximum deviation of 20 m is taken into account.

Robustness.

The maximum differences between the lateral deviation of the trajectories with nominal and perturbed centre of gravity and nominal and maximal time delays are considered:

$$\Delta_{eyb}(t) := \max_t\left(|e_{ybmax}(t)-e_{yb}(t)|,|e_{ybmin}(t)-e_{yb}(t)|\right) \tag{3.3}$$

We will allow differences of 10% of the maximal allowable lateral deviations:

$$\frac{1}{2}\max_t\left(\frac{\Delta_{eyb}(t)}{10}+\frac{\Delta_{eyb}(t_1)}{2}\right) \tag{3.4}$$

should be smaller than one.

Ride quality.

The maximum lateral acceleration will be considered:

$$\max_t\left(\frac{|n_y(t)|}{0.2}\right)<1 \tag{3.5}$$

i.e. $|n_y|$ should be smaller than $0.2g$: under normal flight conditions this value should be much lower ($0.02g$, see section 3.2.4), but engine failure is an emergency situation such that an unusually large lateral acceleration is acceptable.

Safety.

During the segment, the maximum angle of attack α will be considered:

$$\max_t\left(\frac{|\alpha(t)|}{12}\right)^3<1 \tag{3.6}$$

This implies we accept $\alpha = 12$ *deg*; the power is taken to stress the fact that $\alpha > 12$ *deg* quickly becomes unacceptable (stall situation).

Control activity.

The rudder actuator effort will be considered that is needed to stabilise the aircraft after engine failure is recovered. This is calculated as:

$$\int_{t_b}^{t_1}\delta_R^2 dt \tag{3.7}$$

with t_b denoting the end of engine failure (corresponds to point **b** in figure 3.5). This value is not 'normalised to one' as it is not clear what bounds can be obtained: it will act as a value for relative comparison of controllers.

3.4.2 Segment II

For segment II we will plot a plan view of the reference trajectory and the four trajectories defined in section 3.3, and then superimpose the bounds given in figure 3.6. Furthermore,

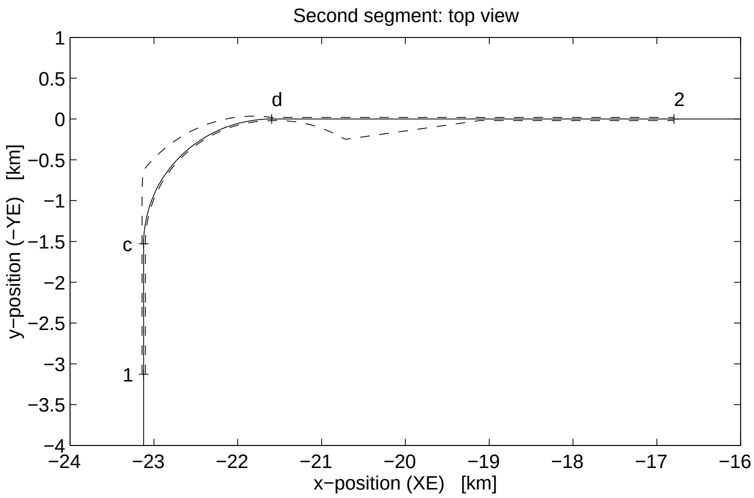

Fig.3.6 Segment II: plan view of the 90-degree turn with bounds

to obtain a better insight in the results, we will plot lateral deviations with bounds as given in figure 3.7.

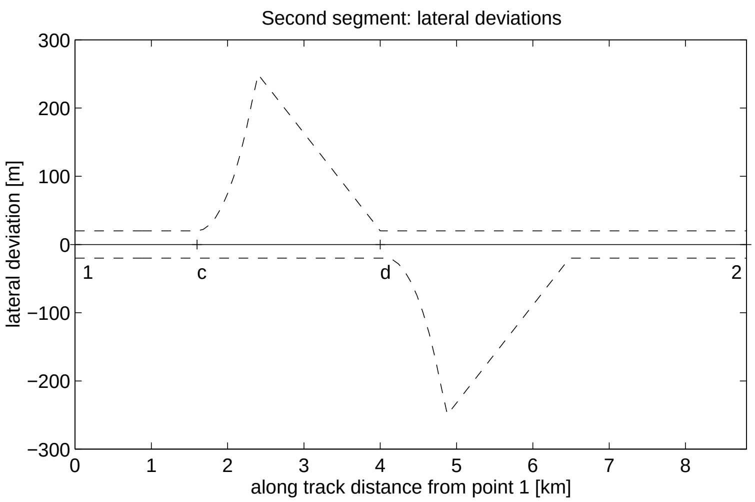

Fig.3.7 Segment II: lateral deviations during the 90-degree turn with bounds

Performance.

The maximum lateral deviation (due to the turn) and the lateral deviation at the end of the segment (when the aircraft should be stabilised again) are considered:

$$\frac{1}{2}\max_t \left(\frac{|e_{yb}(t)|}{200} + \frac{|e_{yb}(t_2)|}{20} \right) < 1 \tag{3.8}$$

Note that during the entire segment a maximum deviation of 200 m should not be exceeded, and that at the end of the segment (t_2 corresponds with point **2**) a maximum deviation of 20 m is taken into account.

Robustness.

As in segment I, the maximum differences between the lateral deviation of the trajectories with nominal and perturbed centre of gravity locations and nominal and maximal time delays are considered. Again, we will allow differences of 10% of the maximal allowable lateral deviations:

$$\frac{1}{2}\max_t \left(\frac{\Delta_{eyb}(t)}{20} + \frac{\Delta_{eyb}(t_2)}{2} \right) < 1 \tag{3.9}$$

Ride quality.

As in segment I, the maximum lateral acceleration n_y will be considered:

$$\max_t \left(\frac{|n_y(t)|}{0.02} \right) < 1 \tag{3.10}$$

i.e. $|n_y|$ should be smaller than 0.02 g.

Safety.

As in segment I, the maximum angle of attack during the segment is limited:

$$\max_t \left(\frac{|\alpha(t)|}{12} \right)^3 < 1 \tag{3.11}$$

Control activity.

The rudder and aileron actuator effort is calculated as:

$$\int_{t_1}^{t_2} \left(\delta_R^2 + \delta_A^2 \right) dt \tag{3.12}$$

This value is not 'normalised to one' as it is not clear what bounds can be obtained: it will act as a value for relative comparison of controllers.

3.4.3 Segment III

For segment III we will plot the altitude deviation as a side view of the four trajectories defined in section 3.3. Figure 3.8 shows the reference trajectory, the start and end points of the segment (points **2** and **3**) and the considered bounds; the command actions are labeled with **e** and **f**. We will also plot the vertical deviation of the trajectories and overlay the bounds shown in figure 3.9.

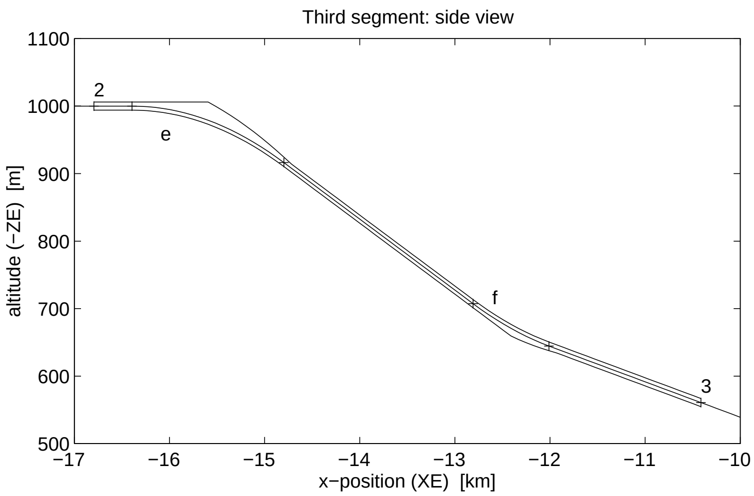

Fig.3.8 Segment III: side view of the -6 and -3 degree glideslope captures with bounds

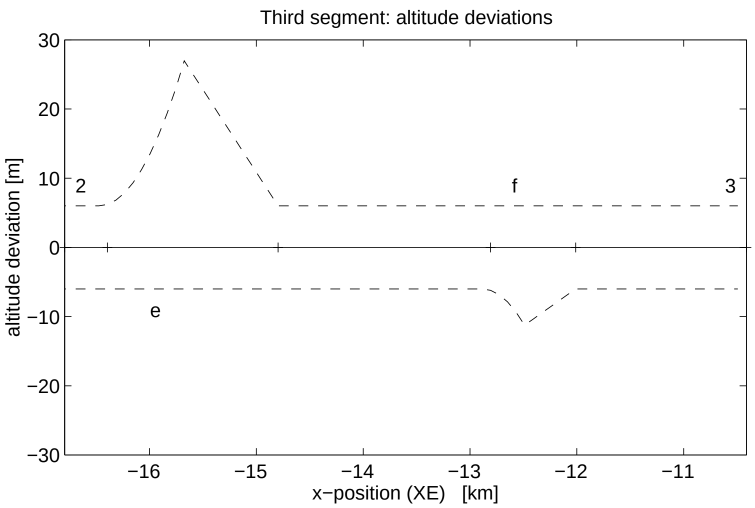

Fig.3.9 Segment III: vertical deviations during the -6 and -3 degree glideslope captures with bounds

Performance.

The maximum vertical deviation during the capture of the -6 degree glideslope and the vertical deviation at the end of the segment (when the aircraft should be stabilised again) are considered. Furthermore, speed variations should be kept small in spite of the change in required angle of attack. With e_{zb} denoting the vertical deviation in body co-ordinates for the trajectory with nominal centre of gravity and time delay, we will demand

$$\frac{1}{3}\max_t\left(\frac{|e_{zb}(t)|}{20}+\frac{|e_{zb}(t_3)|}{6}+\frac{|V_A-V_{Ac}|}{4}\right)<1 \tag{3.13}$$

for sufficient performance. Note that during the entire segment a maximum deviation of 20 m should not be exceeded, and that at the end of the segment (t_3 corresponds with point **3**) a maximum deviation of 6 m is taken into account. Speed variations should not exceed 4 m/s, i.e. 5% of $V_{Ac} = 80\ m/s$).

Robustness.

The maximum differences between the vertical deviation of the trajectories for the nominal and perturbed centre of gravity locations and nominal and maximal time delays are considered:

$$\Delta_{ezb}(t) := \max\left(|e_{zbmax}(t)-e_{zb}(t)|, |e_{zbmin}(t)-e_{zb}(t)|\right) \tag{3.14}$$

We will allow differences of 10% of the maximal allowable vertical deviations:

$$\frac{1}{2}\left(\max_t\frac{\Delta_{ezb}(t)}{2}+\frac{\Delta_{ezb}(t_3)}{0.6}\right)<1 \tag{3.15}$$

Ride quality.

The maximum vertical acceleration n_z will be limited:

$$\max_t\left(\frac{|n_z(t)|}{0.05}\right)<1 \tag{3.16}$$

i.e. $|n_z|$ should be smaller than 0.05 g.

Safety.

Again, the maximum angle of attack during the segment is limited:

$$\max_t\left(\frac{|\alpha(t)|}{12}\right)^3<1 \tag{3.17}$$

Control activity.

The tailplane actuator effort is calculated as:

$$\int_{t_2}^{t_3}\delta_T^2\,dt \tag{3.18}$$

This value is not 'normalised to one' as it is not clear what bounds can be obtained: it will act as a value for relative comparison of controllers.

3.4.4 Segment IV

For segment IV we will plot a side view of the four trajectories defined in section 3.3. The wind shear model, the desired trajectory through it, and the bounds are given in figure 3.10. The wind shear model is a two-dimensional model derived from a three-

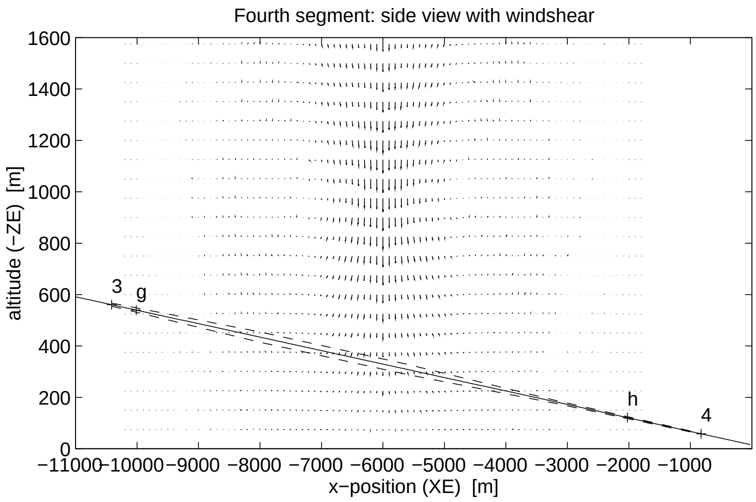

Fig.3.10 Segment IV: side view of the final approach with wind shear and bounds

dimensional model [13]. Along the trajectory, the aircraft will be faced with a headwind going up to about $W_{XE} = -7\ m/s$, then wind speed will change to a tailwind of about $W_{XE} = 7\ m/s$, combined with a downdraught of about $W_{ZE} = 8\ m/s$ (see figure 3.11). The result of this will be a drastic decrease in aircraft energy: the aircraft will not be able

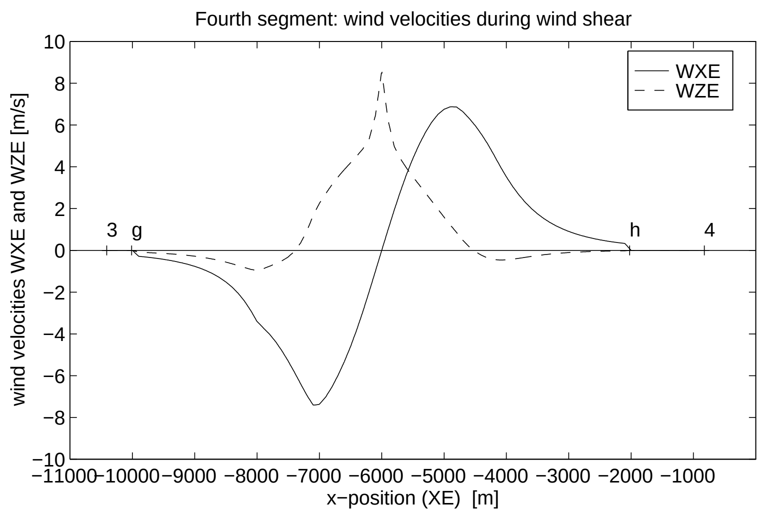

Fig.3.11 Segment IV: wind speeds along the trajectory

to stay on the desired trajectory. The size of the longitudinal deviation and the time until

recovery will be measures for evaluation of the controller.
For this reason we will also plot the longitudinal deviations with bounds as given in figure 3.12.

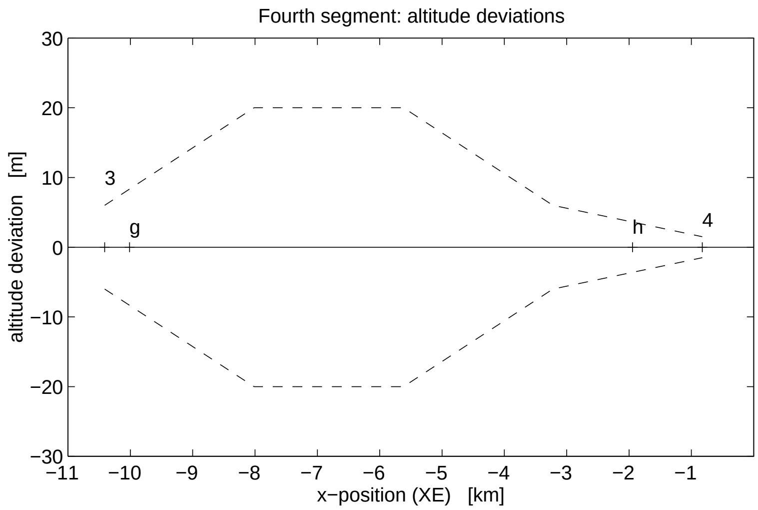

Fig.3.12 Segment IV: vertical deviations during the final approach with bounds

Performance.

The maximum longitudinal deviation (due to the wind shear) and the longitudinal deviation at the end of the segment (when the aircraft should be within the decision window) are considered for the trajectory with nominal centre of gravity and time delay:

$$\frac{1}{2}\max_t\left(\frac{|e_{zb}(t)|}{20}+\frac{|e_{zb}(t_4)|}{1.5}\right)<1 \qquad (3.19)$$

Note that during the entire segment a maximum deviation of 20 m should not be exceeded, and that at the end of the segment (t_4 corresponds with point **4**) a maximum deviation of 1.5 m is taken into account.

Robustness.

As in segment III, the maximum differences between the vertical deviation of the trajectories for the nominal and perturbed centre of gravity locations and nominal and maximal time delays are considered. Again, we will allow differences of 10% of the maximal allowable vertical deviations:

$$\frac{1}{2}\max_t\left(\frac{\Delta_{ezb}(t)}{2}+\frac{\Delta_{ezb}(t_4)}{0.15}\right)<1 \qquad (3.20)$$

Ride quality.

As in segment III, the maximum vertical acceleration n_z will be considered:

$$\max_t\left(\frac{|n_z|}{0.2}\right)<1 \qquad (3.21)$$

i.e. $|n_z|$ should be smaller than $0.2g$ (in segment III this value is lower, but wind shear is an extreme situation).

Safety.

We will consider whether the aircraft is within the decision window at the end of the segment. Lateral, vertical and speed variations are limited to 5 m, 1.5 m and 3 m/s respectively as follows:

$$\sqrt{\frac{1}{3}\left((\frac{e_{yb}}{5})^2+(\frac{e_{zb}}{1.5})^2+(\frac{(V_A-V_{Ac})}{3})^2\right)}<1 \qquad (3.22)$$

Control activity.

The tailplane and throttle actuators effort is calculated as follows:

$$\int_{t_3}^{t_4}\left(\delta_T^2+(\delta_{TH_1}+\delta_{TH_2})^2\right)dt \qquad (3.23)$$

This value is not 'normalised to one' as it is not clear what bounds can be obtained: it will act as a value for relative comparison of controllers.

4 Design entry document layout

4.1 Introduction

The objective of this chapter is to provide guidelines to the participants on how to present their results to the Action Group. It is intended that this will result in a uniform presentation of the results from each of the participants despite the use of a wide variety of methodologies and will hence make comparisons much easier. The main aim of the design challenge is not just to obtain 'good' or even 'excellent' controllers: as mentioned before, the given design problem has already been solved for many similar aircraft. The purpose of the design challenge is to obtain insight into the relative merits of several design methodologies. Therefore, it is stressed that the contributions should be tutorial in nature: this implies that it must be possible to retrace the applied procedure and independently redesign the resulting controller(s). Furthermore, it is considered of great importance that all necessary assumptions and design objectives are well motivated and related to the general design specifications given in section 3.2. The suggested layout and structure of the standard presentation format is intended to filter out the specific design aspects relevant for each method, such that a clear idea about the performance of each design method is obtained. The performance of a method will be assessed in terms of flexibility, applicability, generality, and effectiveness, thereby providing economic guidelines for the industry and research institutes.
The automated evaluation procedure for the resulting controller as described in section 3.3 is only a part of the final evaluation of the reported design methodology. More specifically, the following aspects should be considered, with approximately equal weight to each of the main items:

- the tutorial value of the entry;
  - the general description of the method,
  - the set up of a controller architecture,
  - the motivation of assumptions made,
  - the motivation for the use of method specific design objectives,
  - the translation of general design specifications into method specific design objectives,
  - the selection of weight functions and trade-off parameters,
  - the execution of the design cycle,
  - the method dependent analysis of results,
- the (estimated) effort necessary for application of the methodology;
  - the complexity of the method,

  - the effort related to the setting up of the design cycle (modelling, controller architecture, weight functions)
  - the effort related to the execution of the design cycle (numerical effort, degree of automation)
  - the effort of performing a redesign after a major aircraft design change,

- the complexity of the control solution;

  - the controller architecture (required measurement signals, reference signals, modes, actuators and filters),
  - non-linearity of the controller (adaptive, gain scheduling),
  - (linear) order of the controller,
  - ease of implementation,

- the behaviour of the controller as found by the automated evaluation procedure of section 3.3.

This implies for instance that information on duration of each design iteration and the motivation for each relevant design action have to be reported. Furthermore, specific problems should be pointed out and discussed.

The structure proposed for the standard document to be prepared by each design challenge contestant is aimed at accommodating all these aspects. The next section will give a short overview of this structure, after which each element will be discussed in more detail. Framework documents that accommodate this structure are available in LaTeX and WordPerfect: if necessary, the correct use of these documents is indicated.

4.2 Standard presentation format layout

In a summarised form, the standard presentation format will consist of a document with the following structure:

- title, table of contents, list of figures, list of tables, list of symbols and abbreviations;
- summary;
- chapter 1: introduction;
- chapter 2: a tutorial review of the applied control design methodology,
  - introduction,
  - typical applications,
  - plant model requirements,
  - controller structure,

  - possible design objectives,
  - design cycle description,
  - a simple design example (optional);
- chapter 3: the selection of the controller architecture for the RCAM problem,
  - required measurement signals,
  - required actuator signals (control effectors),
  - required filters, (reference) models,
  - required reference signals;
- chapter 4: the translation of RCAM design criteria into method dependent objectives, for instance (if applicable):
  - time domain criteria into frequency domain criteria,
  - time domain criteria into pole-zero criteria,
  - the definition of cost functions,
  - the setting up of an interconnection structure,
  - graphical methods,
  - non-linear specifications into linear specifications,
  - etc;
- chapter 5: the description of the design cycle,
  - required numerical tools for controller synthesis/analysis,
  - intermediate analysis,
  - design parameter adjustment strategy;
- chapter 6: analysis of the resulting controller in terms of the applied methodology, for instance (if applicable):
  - closed loop frequency domain analysis,
  - open loop frequency domain analysis at actuators and sensors: gain and phase margins, roll off actuator loop,
  - singular value or structured singular value analysis,
  - covariance response or RMS response of states and control signals to disturbances and gusts,
  - robust performance assessment,
  - time domain simulations: linear and non-linear;
- chapter 7: results of the automated evaluation procedure;

- chapter 8: conclusions;
- references;
- appendices, etc:
  - Appendix A: Used software;
  - Appendix B: Background information on people involved creating the entry;
  - Appendix C: Results of the RCAM Assessment software (see Appendix D);
  - Appendix D: Self-evaluation based on teh Evaluation Questionnaire (see Appendix E);
  - Other extra information can be included in subsequent appendices.

In general, each of the aforementioned main items will give rise to a separate chapter: in the following sections the possible contents of these chapters will be discussed.

4.3 Required contents fo the Design Report

Title page and preamble
The available standard documents are self explanatory with respect to generation of title page and preamble.

Summary
The summary should provide a short description of the applied methodology, the obtained controller and some general comments on the achieved results.

Introduction (chapter 1)
The introduction may be used to provide some information on your organisation, your interest in contributing to the design challenge, and the motivation for the use of the control design methodology you will apply. It should contain a concise problem formulation and an overview of the document's contents.

A tutorial review of the applied control design methodology (chapter 2)
This should explain the aim of your chosen method and its potential, clearly formulating the objectives. You might use a combination of methods for each specific objective and if so, explain why this approach has been adopted and how it achieves the objectives. If the method has some particular features, such as special analysis and synthesis features, these should be described. If the method, a priori, takes into account performance and robustness specifications this should be stated and explained. Does the methodology require gain scheduling? Can the method decouple interaction in loops? Finally, can the method handle feedforward paths, or do you need to consider regulation and feedforward loops separately? It is also important to describe whether the controller is robust in a

linear, or a non-linear sense. Some discussion should also be included about whether the method guarantees stability, eg. consider non-linear or adaptive controllers.

The selection of the controller architecture for the RCAM problem (chapter 3)

Define the control system architecture for the overall system. This means that a description has to be given of the subcomponents in your control system and that the arrangement has to be reasoned. You might choose a uniform and reduced set of variables to command inner loop variables for any selected mode. All this boils down to a functional description of the control system.

Describe the controller structure you have adopted for the design task. For instance, this could be a feedback controller in combination with a feedforward controller for which the design could consist of either separate or simultaneous design of feedforward and feedback loops. Important information on the feedback design is the choice of regulated variables, the use of additional integrators, the use of full or partial state estimation, etc. When considering feedforward design, subjects like performance features, ideal model response, decoupling features and co-ordination can be discussed.

The translation of the RCAM design criteria into method dependent objectives (chapter 4)

The RCAM design criteria are set up in method independent terms in section 3.2. This chapter should consider these requirements and provide a motivated procedure to approximate them by means of objectives that are of significance for the proposed design methodology. A discussion may be given with respect to the specific properties of the possible design method dependent objectives and their potential to reflect the given requirements: it is to be expected that some requirements allow a good representation, while others are much harder to incorporate. Indicate your opinion on the application area of the method.

The description of the design cycle (chapter 5)

This chapter should consider the numerical tools and methods necessary to perform the actual design cycle. A description should be given of the necessary actions that are to be taken for each iteration. An important aspect is, for instance, whether it is possible to automate the procedure and to what extent expert knowledge of the designer is required for intermediate decisions. This implies an extensive description of weight function selection criteria, design parameters and search strategies as well as a discussion on the convergence of the iteration procedure.

Analysis of the resulting controller in terms of the applied methodology (chapter 6)

The analysis of the resulting design will be dependent on the applied methodology. It should be made clear to what extent the controller satisfies the design objectives formulated in chapter 4 of the Design Report. Again, it is necessary to consider the relation between the design objectives and the original design requirements formulated in section 3.2. Methods and indicators that may be of interest for a specific design method could be:

- eigenvalues, minimum damping,
- broken loop frequency analysis at actuators and sensors, gain margins and phase margins, actuator loop roll off,
- singular value and structured singular value analysis,
- covariance response, RMS values of state variables and control inputs for given disturbances and gusts,
- ride quality indicators,
- control activity indicators,
- robust performance indicators,
- cost functions,
- linear and non-linear simulations.

Results of the automated evaluation procedure (chapter 7)
The control design method independent evaluation procedure is fully automated and will result in a single chapter in the final document. Default text is provided but may be adapted to comment on particular results. It is preferred that no changes are made to any of the automatically generated figures or the table with numerical results. If a given figure does not satisfactorily represent the controller's behaviour, please do not remove it: if desired an extra figure may be added to explain your point.

Conclusions (chapter 8)
This chapter should comment on overall aspects of the design methodology and its features with respect the RCAM design problem. Specific strong and weak points of the method should be discussed. Possible future extensions or improvements of the presented method should be indicated. Some comments should be given on the performance of the method related to the four main objectives mentioned in section 4.1. Any comments on the set up of the design challenge or the considered design problem can also be considered here.

References, Appendices, etc.
For the style of referencing see this document.

Appendices containing extra information, for instance on successful applications, required and/or available software and software generated for the purpose of the application of the method to the RCAM design challenge (Matlab m-files, etc.) may be added if desired.
Note that it is the intention to be able to reperform the presented design method. All software and information necessary for this purpose should be documented or made available (by means of ftp etc.). This information should be supplied in Appendix A.
Appendix B should contain background information on the people involved in the design and creation of the entry. It would be especially interesting to know something about previous experience in flight mechanics and flight control problems as well as the application of the described method.

4.4 Final remarks

As mentioned before, standard document layouts will be available in WordPerfect and LaTeX. These documents can be used as a framework for your design challenge entry document. Default texts are inserted when possible but may be adjusted to your specific needs. Note that the layout is chosen to accommodate a large number of methodologies and is aimed at an objective and complete comparison. Any comments you may have on these standard layouts and the entire procedure are welcomed.

References

[1] GARTEUR Exploratory Group FM-EG12, *Terms of Reference for GARTEUR Flight Mechanics Action Group (FM-AG08): Robust Flight Control in a Computational Aircraft Control Egineering Environment*, National Aerospace Laboratory (NLR), Amsterdam, January, 1995.

[2] GARTEUR Action Group FM(AG08), *Communication Handbook*, Version 2, GARTEUR/TP-088-5, To appear: February, 1996.

[3] GARTEUR Action Group FM(AG08), *RCAM Preliminary Design Document*, GARTEUR/TP-088-9, November 1995.

[4] Brockhaus R., *Flugregelung*, Springer Verlag, Berlin, 1994.

[5] Brumbaugh R. W., 'Aircraft Model for the AIAA Controls Design Challenge', *Journal of Guidance, Control and Dynamics*, Vol.17, No.4, 1994.

[6] Dorato P. (ed.), *Robust Control*, IEEE Press, 1987.

[7] Dorato P. and Yedavalli R.K. (eds.), *Recent Advances in Robust Control*, IEEE Press, 1990.

[8] Etkin B., *Dynamics Of Flight - Stability and Control*, John Wiley & Sons, USA, November 1958.

[9] Favre C., 'Fly-by-wire for commercial aircraft: the Airbus experience', *Int. J. Control, Special Issue on Aircraft Flight Control*, Vol.59, No.1, pp.139-158, 1994.

[10] Franklin G.F., Powell J.D. and Emami-Naeni A., *Feedback Control of Dynamic Systems*, Addison-Wesley 1986.

[11] Kaminer I., Benson R.A., Coleman E.E. and Ebrahimi Y.S., 'Design of Integrated Pitch Axis for Autopilot/Autothrottle and Integrated Lateral Axis for Autopilot/Yaw Damper for NASA TSRV Airplane Using Integral LQG Methodology', *NASA CR-4268*, January, 1990.

[12] McRuer, D., 'Interdisciplinary interactions and dynamic systems integration', *Int. J. Control, Special Issue on Aircraft Flight Control*, Vol.59, No.1, Jan. 1994.

[13] Robinson P.A., 'The Modelling of Turbulence and Downbursts for Flight Simulators', *UTIAS report No.339*, University of Toronto, 1991.

[14] Wise K. et al., 'Linear and Nonlinear Aircraft Flight Control for the AIAA Control Design Challenge', *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 1992.

[15] Ying-Jyi P.W., 'Intelligent Control Law Tuning for AIAA Control Design Challenge', *Journal of Guidance, Control and Dynamics*, Vol.17, No.4, 1994.

[16] *Matlab User's Guide*, The MathWorks Inc., 24 Prime Park Way, Natick, Mass. 01760, USA, July 1993.

[17] *Simulink User's Guide*, The MathWorks Inc., 24 Prime Park Way, Natick, Mass. 01760, USA, April 1993.

[18] *MATRIX$_X$/SystemBuild V.2.4, User's Guide*, Integrated Systems Inc., Santa Clara, CA, USA, 1991.

[19] R. Brockhaus. *Flugregelung.* Springer Verlag, Berlin Heidelberg, 1994.

[20] F.E. Cellier. *Continuous System Modeling.* Springer-Verlag, New York, 1991.

[21] H. Elmqvist. *Dymola – Dynamic Modeling Language User's Manual.* Dynasim AB, S–22370 Lund, Sweden, 1993.

[22] H. Elmqvist. Object-Oriented Modeling and Automatic Formula Manipulation in Dymola. In *Scandinavian Simulation Society SIMS'93*, Kongsberg, Norway, June 1993.

[23] B. Etkin. *Dynamics of Flight - Stability and Control.* John Wiley & Sons, USA, November 1958.

[24] G. Grübel, D. Joos H, M. Otter, and R. Finsterwalder. The ANDECS Design Environment for Control Engineering. In *12th IFAC World Congress, Sydney, Australia, July 19-23*, volume 6, pages 447–454, 1993.

[25] A. Helmersson, K. Ståhl Gunnarsson, J. Terlouw, and P. Lambrechts. Communication Handbook. *by GARTEUR Flight Mechancis Action Group 08*, March 11, 1996.

[26] P. F. Lambrechts and J. C. Terlouw. A Matlab Toolbox for Parametric Uncertainty Modeling. *Philips Research Eindhoven and NLR Amsterdam*, 1992.

[27] M. Otter. *Objectorientierte Modellierung mechatronischer Systeme am Beispiel geregelter Roboter.* VDI Fortschrittsberichte, Reihe 20: Rechnerunterstützte Verfahren, 1995.

[28] M. Otter. Dsblock: A neutral description of dynamic systems. version 3.2. In *Technical Report TR R81-92, DLR, Institute for Robotics and Systemdynamics, Oberpfaffenhofen, D-82234 Wessling, Germany*, May 1992.

[29] M. Otter, H. Elmqvist, and F.E. Cellier. Modeling of multibody systems with the object-oriented modeling language dymola. *Nonlinear Dynamics 9, 1996 Kluwer Academic Publishers. Printed in the Netherlands*, pages 91–112, 1996.

[30] W. Sienel, T. Bünte, and J. Ackermann. Paradise – a Matlab-based robust control toolbox. In *Proc. IEEE Symposium on Computer-Aided Control System Design*, Dearborn, MI, 1996.

[31] Flying Qualities of Piloted Aircraft. MIL-STD-1797A, 1990.

A The RCAM model and design environment software description

In this chapter the software for the six degree-of-freedom RCAM model, as detailed in chapter 2, is described.
The software code is automatically generated by Dymola, where the objects, given in figure 2.2, are coded in the form of equations. The connections between the objects represent the physical interactions. From the physical description set up in Dymola, a consistent symbolical mathematical model is built automatically by the Dymola symbolic equations handler. This mathematical model is then used to generate efficient simulation code for different simulation environments.
For the Matlab/Simulink simulation environment, code can be generated in the Matlab m-file and/or mex-files (Fortran or C) formats.
Also Fortran or C-code according to the neutral DSblock format may be generated, which can be directly used within the ANDECS simulation environment.
The RCAM software supplied with this manual uses the C-code version of the RCAM model; it is relatively easy to use and much faster than the m-file version. Compiled code for use on IBM-compatible PCs is included with the software and can be used immediately. For compilation with your own C-compiler or on workstations you can use the `cmex` batch-file supplied with Matlab* (see section A.1).
If you are interested in other versions of the RCAM software model, see section A.5.

A.1 Installation

We assume that you have a correctly installed version of Matlab/Simulink (Matlab version 4.2 or higher, Simulink version 1.3c or higher) on a workstation or a, preferably fast, PC. As mentioned before, on a workstation you also need a C-compiler that can be used in combination with `cmex`; on a PC you can use the supplied code, although availability of a C-compiler is recommended.

All files, which are required for the design of the controller should be arranged in a single directory, for instance:

```
....\GARTEUR\RCAM\RCAM-DES
```

You can obtain these files from anonymous ftp.
The following procedure should be executed for installation:

- create on your hard disk a new directory to work in and make this your current directory (e.g. `./garteur`),
- start ftp; check whether your current local directory is still your intended work directory,

*You should consult the Matlab manuals for a list of compatible C-compilers. For PCs, the use of WATCOM 9.0, 9.5, 10.0, or 10.5 is recommended.

- from the ftp> prompt: enter `open ftp.nlr.nl`, enter username `anonymous` and supply you e-mail address as password,
- change remote directory by entering the command: `cd transit`
- do not be alarmed when the `ls` command reports that no files are available: this is done for security reasons,
- make sure that the file transfer mode is set to binary by entering the command: `bin`,
- now get the file `rcam3721.uue` by entering the command: `get rcam3721.uue`,
- leave ftp and check whether your current directory is still your intended work directory,
- decode, uncompress and untar the file `rcam3721.uue`:
  ```
  uudecode rcam3721.uue
  uncompress -f rcam.tar.Z
  tar xvfo rcam.tar,
  ```
- For use on a workstation, go to directory: `./rcam/rcam-des`
  then compile the cmex-file `rcamex.c` by entering the command:
  `cmex rcamex.c`,

A.2 Installed files

After successful installation, you should have the following files:

```
trimrcam.m   rcam9.m
rcam_des.m
control.m    control.mat
rcamex.c    (rcamex.mex rcamex.obj)
init.mat
```

The routine `trimrcam.m` is used together with `rcam9.m` to set the initial conditions of the model. `rcam_des.m` is the Simulink design environment to be used for non-linear simulations with a designed controller. As an example, the controller resulting from the preliminary design study [3] is supplied in `control.m`, with its corresponding data in `control.mat`. The S-function that describes the RCAM dynamics in C source code is given as `rcamex.c` (`rcamex.mex` and `rcamex.obj` are compiled files for use on a PC). Finally, the file `init.mat` contains all variables (controller, parameters, etc.) necessary for simulation (this file is automatically generated if you use `trimrcam`).

A.3 The design environment

The design environment is intended to be a tool in the development of your own controller. You can start it by entering `rcam_des` at the Matlab prompt, after which you should get the window as given in figure A.1. Within the design environment you are free to vary

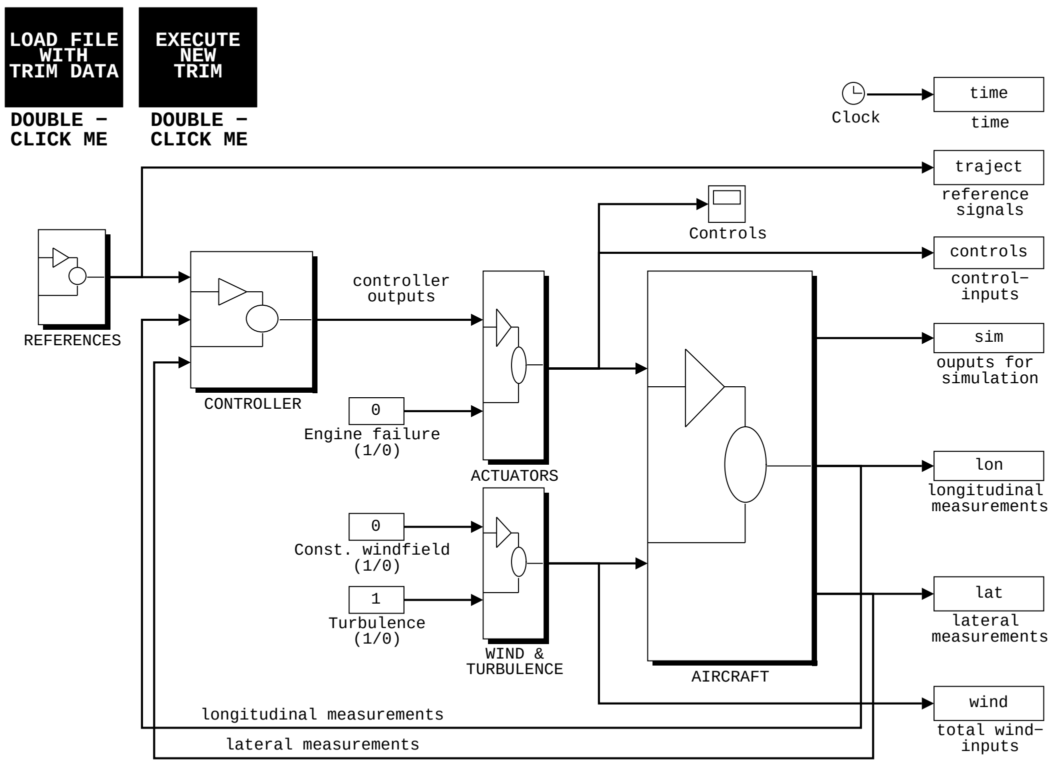

Fig.A.1 Simulink design model `rcam_des.m`

the structure and format of the models as desired. However, it is **essential** that the final design of the controller works correctly with the original models, as they are used in the evaluation procedure. For the same reason, the controller should have exactly the same number and order of inputs and outputs as used in the original environment, and as indicated in figure A.2. The best way to ensure compatibility of your controller with the design environment and the evaluation procedure is to start with the given example controller in `control.m` (you can open it by entering `control` at the Matlab prompt), and to replace the appropriate blocks with your own controller.

Note that you can define your controller in many ways: you can build it using standard Simulink blocks, you can make use of state-space blocks and transfer function blocks to define linear controllers, you can create S-functions in C, Fortran or Matlab script files, and any combinations. Furthermore, you can make use of the file `control.mat` to specify

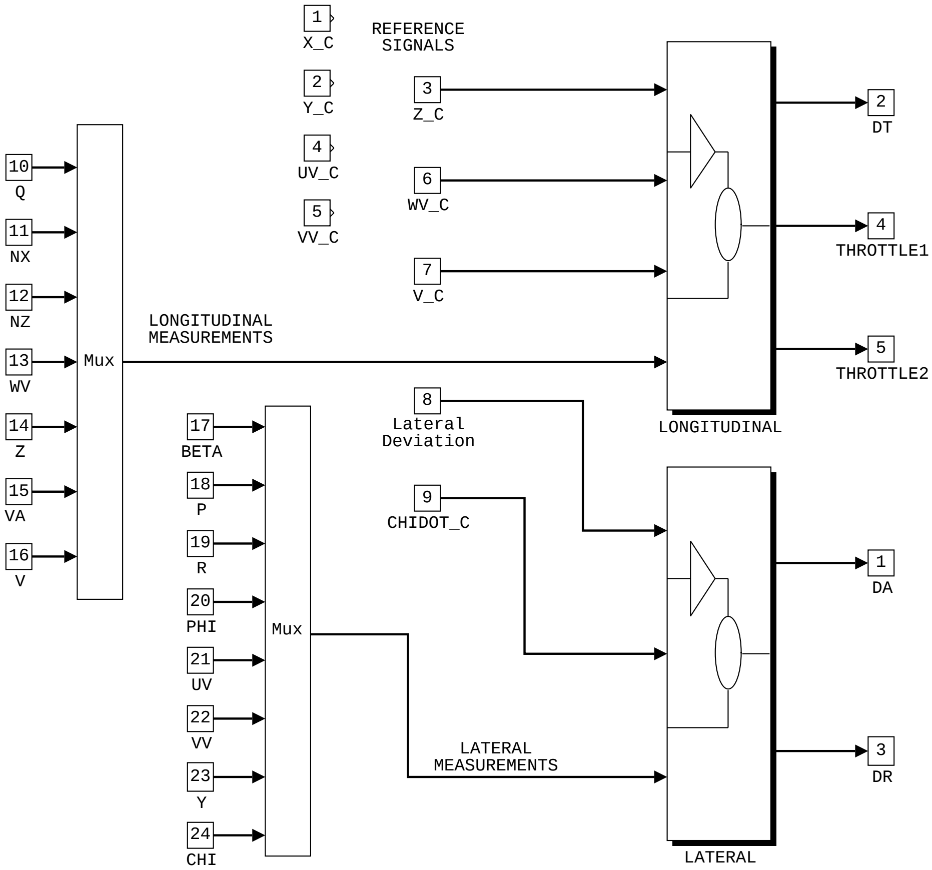

Fig.A.2 Simulink example controller model `control.m`. *The reference signals X_C, Y_C, U_{VC}, V_{VC}, Z_C, W_{VC} and V_C are defining a consistent trajectory. This mean, for instance, that Z_C is the integral of W_{VC}. A command for change in altitude, Z_C can thus be realized as a ramp in Z_C and a step of finite length in W_{VC}.*

data in the form of global Matlab variables, to be used as parameters in your controller (gains, $ABCD$ matrices, etc.).

Finally, note that the block `references` does not contain the trajectory generator needed for the evaluation procedure. It is intended to be used for the analysis of your controller in the sense of the given design specifications and in terms of the applied controller design method.

A.4 Trimming and linearisation

Before you start simulations within Simulink, the RCAM model has to be trimmed. This is achieved by executing `trimrcam.m`, either from the Matlab command prompt or by double-clicking the `execute new trim` block from the design environment. You will be asked to enter the desired parameter values for the new trim condition: the procedure is

self-explanatory and default values are taken from the `init.mat` file.
The result of the trimming procedure will be the initial state vector of the system (`X0`), the initial input vector (`U0`), the initial output vector (`Y0`), and the parameter vector (`p`), which will all be used in the simulation within Simulink.
Additionally `trimrcam` determines a linearisation of the model for this trim condition. The resulting A, B, C, and D matrices of the linear system are computed and stored in the workspace. After that, it is possible to automatically generate some significant open loop responses starting from this trim condition, both using the non-linear model and the linear model. The generated plots give an impression of the quality and scope of the linear model: the numerical results are saved to `sim.mat`.
All these initial vectors, systems matrices and system parameters are saved to `init.mat`. All necessary data for a simulation run within the Simulink design environment can be retrieved by loading `init.mat` and `control.mat` (the latter contains any relevant control parameters). This can be done from within the Simulink design environment by double clicking the `load file with trim data` block).

A.5 Other RCAM model software formats

The model of the RCAM dynamics can also be supplied in several alternative forms:

- a Matlab/Simulink S-function in m-file format,
- a Matlab/Simulink S-function in Fortran code,
- ANDECS-DSblock code,
- 'plain' Fortran or C code,
- the symbolic mathematical model of RCAM in Dymola.
- the model of RCAM in the form of a linear fractional transformation (LFT) for post-design μ-analysis.

To obtain any of these alternatives, contact:
Dieter Moormann, DLR
via e-mail: Dieter.Moormann@dlr.de
or Tel: +49 8153 28 2428 / Fax: +49 8153 28 1441.

B The standard design challenge entry document layout

This manual is provided with a framework document that can be used as a starting point for your design challenge entry document. It is set up in LaTeX and set in the GARTEUR style that we would like you to use (this manual is also set in this style). We strongly advise the use of LaTeX: it is public domain, it runs on many different platforms, and it is well accepted in academia. If you are committed to a different word processor, or if you have any other problems with the software described here, please contact:
Paul Lambrechts, NLR
via e-mail: lambo@nlr.nl
or Tel: +31 20 511 3740 / Fax: +31 20 511 3210.

B.1 Installation

We assume you have a correctly installed version of LaTeX on a workstation or PC (for instance EMTeX) and the possibility to make use of a Postscript printer (or Ghostscript). All files, which are required for the creation of your design challenge entry document can be arranged into a single directory, for which we suggest:

```
....\GARTEUR\RCAM\RCAM-FRA
```

Similar to the RCAM model and design environment software as described in appendix A, you can obtain these files from anonymous ftp.
After installation of the design environment the framework document is in directory `./rcam/rcam-fra`.

B.2 The first test

To check correct transfer of all files and correct operation of your version of LaTeX, it is possible to immediately test whether the document can be compiled and printed:

- go to your intended work directory,
- run LaTeX on the file `rcam-fra.tex`,
- after compilation, run `DVIPS` on the file `rcam-fra.dvi`,
- print `rcam-fra.ps` to your Postscript printer (or use Ghostscript).

B.3 The use of `.sty` files

After successful completion of the first test, you may consider a more permanent installation of the provided software. For this you should locate the subdirectory in which your implementation of LaTeX stores its style files: usually this is the subdirectory `....\TEXINPUT`.

You may also consider a separate subdirectory for the provided `.sty` files, as long as LaTeX knows where to find them. Next, move all `.sty` files to this subdirectory.
Most of the style files are standard, like `bk11.sty` and `epsf.sty`: they are included for completeness. Two of them are specially designed for the GARTEUR FMAG-08 group: `garteur.sty` and `fmag.sty`. `garteur.sty` replaces standard style files like `book.sty` and `article.sty`; `fmag.sty` is used for some additional definitions. See `rcam-fra.tex` for more information on the use of these style files.
The two Encapsulated Postscript files `garteur.eps` and `garthead.eps` must remain in the same directory as `rcam-fra.tex`.

B.4 The example files

To make sure that the framework document `rcam-fra.tex` can be compiled, we have added some example files that result from the automatic evaluation procedure described in appendix C. These files were generated using the example controller discussed in appendix A (`control.m` and `control.mat`). The files consist of a number of Encapsulated Postscript files (with extension `.eps`) and the files `rcam-tbl.tex` and `rcam-tbl.txt`, which are also automatically generated and contain the numerical results of the evaluation procedure (`.tex` is prepared for LaTeX use, `.txt` is a simple ASCII file with the same results).

C The automated evaluation software

The automated evaluation software can be used to evaluate any controller designed with the help of the design environment discussed in appendix A. It is the intention that you design your controller within the design environment, and that you also use this design environment to apply any evaluation techniques that you prefer to show that your controller meets the design objectives. The evaluation software should only be used for the automated evaluation procedure, to produce the results needed for an objective comparison of different control design methods.

C.1 Installation

We assume that you have successfully installed the RCAM model and design environment software described in appendix A. Similar to this software, you should place all files required for the automated evaluation in a single directory, for instance:

```
....\GARTEUR\RCAM\RCAM-EVA
```

You can obtain these files from anonymous ftp.
After installation of the design environment the evaluation environment can be found in directory `./rcam/rcam-eva`. For use on a workstation, you should compile the cmex-files `rcamex.c` and `traject.c` by entering the commands:

```
cmex rcamex.c
cmex traject.c
```

You should now have at least the following files:

```
evaluate.m  evalplot.m   rcam_tbl.m
rcam_eva.m  tdisplay.m
traject.c   (traject.mex traject.obj)
control.m   control.mat
rcamex.c    (rcamex.mex  rcamex.obj)
init.mat    trim1.mat    trim2.mat    trim3.mat    trim4.mat
```

Note that some files supplied with the automated evaluation procedure are equal to files supplied with the design environment: the S-function of the RCAM dynamics `rcamex.c` and the example controller `control.m` and `control.mat`. None of these files should be changed, except `control.m` and `control.mat`, which contain the definition of the controller to be evaluated.

C.2 The first test

The example controller (see [3]) is supplied with the evaluation environment to check whether all files are properly installed:

- start Matlab,
- change directory to the one in which you installed the automated evaluation software: e.g. `GARTEUR\RCAM\RCAM-EVA`,
- run the Matlab script file `evaluate.m`: ≫ `evaluate` [return],
- the results are now automatically generated: with the example controller this takes about 10 minutes on a 90 MHz Pentium PC; you can check the progress in the Matlab command window: there are four runs taking 500 to 1000 simulation seconds,
- plots are shown: press a key to go through them,
- some ASCII files are created: you see them scroll up in the Matlab command window,
- successful completion is indicated.

The results are now available in your work directory. To incorporate them into the framework document you should copy the following files to the directory in which you have your framework document (see appendix B):

- all `.eps` files,
- `rcam-tbl.tex`.

Note that you will overwrite some of the files in the target directory.
Next, go to the directory with your framework document and use LaTeX to compile `rcam-fra.tex`: the new results are automatically incorporated. The file `rcam-tbl.txt` is created for use with other word processors.

C.3 Use with your own controller

The controller used for the first test is an S-function named `control.m`. To use your own controller, simply replace this S-function with the one you created with the design environment. If you use the design environment as indicated, your controller should be an S-function with the correct number and order of inputs and outputs for use with the evaluation environment. The procedure for the evaluation is then as follows:

- save the controller S-function you created with the design environment as `control.m`
- save any parameters you need to define this controller in `control.mat`; the following names should NOT be used as they contain evaluation procedure parameters:
  `U0, X0, Y0, delay, p, run, gust, v0, wspeed, wx, wz, x, z`
- copy `control.m` and `control.mat` to your evaluation software subdirectory (e.g. `GARTEUR\RCAM\RCAM-EVA`); overwrite the `control.m` file already existing,
- proceed with the procedure given in the previous section

C.4 The trajectory generator (`traject.c`)

The trajectory generator is a bit different from the way trajectories are generally considered in flight simulation. However, it provides an extension that may be useful in the future. Rather than on geometrical information, the trajectory generator is primarily based on time. In other words, instead of the more commonly used 'along the track variable', time is used as an independent variable for parametrising the trajectory. Obviously, this only works correctly when we assume that the trajectory is followed with a predefined nominal velocity: otherwise there would be no fixed relation between time instant and the exact geometrical position that the aircraft should be in (in the earth-fixed reference frame).

Hence, the trajectory is given as the sequence of positions the aircraft would have in time, with nominal speeds, no dynamics, and no disturbances. t=0 could be defined as the nominal time instant the aircraft should pass the threshold; in `traject.c` it is the starting point of the trajectory: it really makes no difference. The point is that the trajectory generator has an internal state: the delay time. Each integration step the actual aircraft position is compared with the desired one, resulting in an 'along the track error' and, in the perpendicular plane, a lateral and a longitudinal error.

Note that the 'along the track error' is only of importance in 4D-navigation, in 3D-navigation it can be used to calculate a 'delta' delay time, i.e. the time necessary to move along the desired trajectory with the desired ground speed until the desired position is reached. The integral of this delta delay time results in the total delay time. Once the aircraft moves away from the desired trajectory (as defined in 4D) due to disturbances etc., the actual simulation time minus the delay time will act as an alternative 'along the track' variable and will give the desired position in a 3D-scenario.

In the evaluation software, the delay time is recorded during the entire approach, and used to modify the evaluation plots such that the comparison of the trajectories really is in space. Hence, even if two trajectories are flown with completely different speeds, the evaluation plots will coincide in space !

The nice thing about this approach in comparison with the conventional approach is that the actual delay time is available as a known quantity (it is the last output of the trajectory signal generator): a controller can make use of it to minimise the delay time (e.g. feed back delay time to desired speed to make up for lost time). Effectively this implies that you can make a smooth transition from 3D-scenarios to 4D-scenarios.

The outputs generated by the trajectory generator are mostly given in the earth-fixed reference frame. The desired position (x_E, y_E, z_E), and the desired velocity vector (u_E, v_E, w_E) make up the first six outputs. The seventh output is the norm of the desired velocity V_E. Note however, that in the 3D-scenario we are looking at, it is allowed to consider this output as the desired airspeed V_A: any delay time resulting from wind inputs or deviations from the desired trajectory is given by the last output of the trajectory generator, and is of no consequence to the evaluation. Output number eight is the lateral deviation from the trajectory, in the plane perpendicular to the trajectory and in the horizontal ($x_E - y_E$) plane. The ninth output is the desired heading angle rate ($\dot{\psi}$). Outputs ten and eleven are logicals to determine the period of engine failure and the period of the occurrence of a constant wind field. Finally, the last output specifies the momentary delay time, i.e., the time between the actual (simulation) time and the time that the aircraft should pass the same point on the trajectory when it would have followed it exactly with nominal speeds.

Note that there is a certain redundancy in the generated signals: they are consistent with each other, and the control designer may choose any combination of these signals, or any quantity that can be derived from them, as reference signals.
To conclude, note that the reference signals are not available to the controller on beforehand. This is not realistic in practice, but it is intentional because we are interested in analysing the feedback behaviour of the designed controller.

D Description of the RCAM Assessment Software (Version 2.0)

D.1 General

The Assessment Software deals with the compliance of a controller designed for RCAM with the "Performance criteria" specified in section 3.2 and the "Robustness criteria" specified in the same section, under nominal conditions and in conditions suggested by the "Evaluation procedure" in section 3.3.

The applicable conditions can be selected interactively during the assessment process. The assessment consists of three subprocesses:

- Trimming process.
- Simulation process.
- Plotting process.

Both the trimming and the simulation use the 12 states non-linear model `rcamex` in a MEX-file format applicable to the local platform. The current version of file `rcamex.c` is that of August 13, 1996. Preceding the first time you start Matlab for the assessment, change directory to the assessment directory (which contains the Assessment Software). Enter the command:

```
cmex rcamex.c
```

at the operating system prompt, to produce the applicable MEX-file, see also Appendix A. Then start Matlab.

D.2 Trimming process

In order to reduce the time consumed during the assessment, it is advised to perform the trimming of RCAM in all selectable operating conditions only once. It will take a number of hours depending on the performance of the platform.
For convenience of the user, a set of trimming results has been included with the Assessment Software.
Only when the trimming results have to be produced again for some reason, the trimming tool needs to be called by entering:

```
trimrcmx
```

at the Matlab prompt. The trimming will be performed for a pre-programmed set of operating conditions:

item	indication	value	case indication	case number
mass	nominal	120 000 kg	mcase	0
	minimum	100 000 kg		1
	maximum	150 000 kg		2
longitudinal CoG	mid	0.23 $\bar{c}$	xcase	0
	forward	0.15 $\bar{c}$		1
	aft	0.31 $\bar{c}$		2
vertical CoG	mid	0.10 $\bar{c}$	zcase	0
	lower	0 $\bar{c}$		1
	upper	0.21 $\bar{c}$		2

and:

item	indication	extra condition	speed	case indication	case number
extended envelope	nominal		1.23 V_{stall}	excase	0
	right engine failure		1.23 V_{stall}		1
	left engine failure		1.23 V_{stall}		2
	turn to right	30° roll	1.32 V_{stall}		3
	turn to left	−30° roll	1.32 V_{stall}		4
	descent −6°		1.23 V_{stall}		5
	max. flap speed		90 m/s		6
	original design speed		80 m/s		7

In total these are $3*3*3*8 = 216$ conditions, for which the trimming process is performed. The results are stored as MAT-files with file names in the format: `rcx****.mat` where `****` indicates the case numbers in the order `mcase xcase zcase excase` .
The aircraft is trimmed at an altitude of 1000 m, without wind. When required, these default values can be changed by editing `trimrcmx.m` .
The trimming uses `ftrim.m` , `limits.m` , `fminv.m` and `aceval.m` , for calculating initial conditions in straight symmetric, straight asymmetric and turning flights.

D.3 Simulation process

In order to assess the performance of a given controller, the files `control.m` and `control.mat` pertaining to that controller shall be copied to the assessment directory, overwriting the files with these names originally present in the Assessment Software, pertaining to a preliminary example controller.

Just before each simulation starts, a user-definable script is called. Its name is `userasm.m` . It has been created to initialize the controller, when needed. Moreover it may be edited by the user, e.g. to change the integration parameter specifications.
The simulations are started by entering:

```
nlasm
```

at the Matlab prompt. The script responds by asking for the wanted type of assessment. In order to produce compact results, in a number of cases two types have been combined in one selection. Enter the applicable number to make a selection:

lateral & altitude step response	(1)
heading & flight-path step response	(2)
roll and heading at engine failure	(3)
airspeed step & wind response	(4)
airspeed - altitude cross coupling	(5)
PSD longitudinal turbulence response	(6)
PSD lateral turbulence response	(7)
PSD vertical turbulence response	(8)

An overview of the generated inputs will be given in section D.4.
Then, the parameter combinations for which the assessment(s) will be performed have to be specified. At each question, any combination of the given possibilities may be given, in any order or number. Default, by hitting only the Enter-key, the possibility indicated by 0 is specified. The possible combinations cover all conditions for which the trimming has been performed, see above, with three possible values of the time delay:

item	indication	value	case indication	case number
time delay	nominal	0.075 s	tdcase	0
	minimum	0.05 s		1
	maximum	0.1 s		2

So a total of $216 * 3 = 648$ different combinations can be selected. Of course, it is advisable to explore some partial sets first, before selecting larger numbers of combinations, which require much more time and memory.
One could start with assessing along the individual axes of the multi-dimensional space of all possibilities, i.e.:

- all assessment selections for the nominal case,
- for the first assessment selection, all time delay cases with all other parameters at their nominal value,

- for that assessment selection, successively assess the influence of all other parameters, one-by-one,
- for that assessment selection, combinations of parameters producing the largest deviations from the nominal case,
- and so on for the other assessment selections.

This will yield a rather dependable indication of the effect of the various parameters. A data base system will keep track of the combinations dealt with, to avoid duplicate simulations.
It has been proposed by the RCAM Evaluation Committee to perform assessment selections 1, 2, 4 and 5 for the following set of 40 combinations:

case	input
time delay	2
mass	12
longitudinal CoG	12
vertical CoG	12
extended envelope	06531

assessment selection 3 for the following set of 32 combinations:

case	input
time delay	2
mass	12
longitudinal CoG	12
vertical CoG	12
extended envelope	0653

and assessment selections 6, 7 and 8 for the following set of 8 combinations:

case	input
time delay	0
mass	12
longitudinal CoG	12
vertical CoG	12
extended envelope	0

During the simulations some information is written to the Matlab command window, to monitor the simulation process, consisting of:

- a counter incrementing at each new parameter combination,
- an identification of the current simulation,
- the value of the time variable during the current simulation.

The simulation process is very time consuming. However, it may be left unattended. Most results are saved with file names in the format: `p*******.mat` where `*******` is a code representing assessment selection, parameter combination and subplot number. It is not advised to manipulate these files directly. The data base system data are stored in a file: `adm*.mat` where `*` is the assessment selection number. The resulting files are used in the interactive plotting process.

During assessments 6, 7 and 8 some results are calculated for presentation in a print file. Those results are saved with file names in the format: `g*******.mat` . The print file will be prepared during the plotting process.

D.4 Input signals

In Table D.1 a list of all elements of the input signal vector is given.

nr.	symbol	description	unit
1	T	time	s
2	X_C	command for x position of aircraft in F_E	m
3	Y_C	command for y position of aircraft in F_E	m
4	Z_C	command for z position of aircraft in F_E	m
5	UV_C	command for x comp. of inert. velocity in F_V ($\parallel F_E$)	m/s
6	VV_C	command for y comp. of inert. velocity in F_V	m/s
7	WV_C	command for z comp. of inert. velocity in F_V	m/s
8	V_C	command for total inertial velocity	m/s
9	DY_C	command for lateral deviation	m
10	PSID_C	command for heading rate	rad/s
11	WXE	x comp. of wind velocity in F_E	m/s
12	WYE	y comp. of wind velocity in F_E	m/s
13	WZE	z comp. of wind velocity in F_E	m/s
14	WXB	x comp. of wind velocity in F_B	m/s
15	WYB	y comp. of wind velocity in F_B	m/s
16	WZB	z comp. of wind velocity in F_B	m/s
17	L E_F	left engine failure	–
18	R E_F	right engine failure	–
19	CTLR_OFF	controller off	–
-	V_0	initial condition of total inertial velocity	m/s

Table D.1 Elements of the input (reference) signal vector

In each assessment some signals will act as an excitation whereas all other elements stay at a zero value, w.r.t. their initial value. In Tables D.2 until D.9 the excitations are presented, as functions of time.

lateral step response						
T	0	1.999	2	3	3.001	40
VV_C	0	0	1	1	0	0
Y_C	0	0	0	1	1	1
DY_C	0	0	0	1	1	1
PSID_C	0	0	$4/V_0$	$-4/V_0$	0	0

altitude step response						
T	0	1.999	2	3	3.001	40
WV_C	0	0	1	1	0	0
Z_C	0	0	0	1	1	1

Table D.2 Assessment selection 1: lateral & altitude step response

heading step response						
T	0	1.999	2	3	3.001	40
VV_C	0	0	1	1	1	1
Y_C	0	0	0	1	1	38
DY_C	0	0	0	1	1	38
PSID_C	0	0	$1/V_0$	$1/V_0$	0	0

flight-path step response						
T	0	1.999	2	3	3.001	40
WV_C	0	0	-1	-1	-1	-1
Z_C	0	0	0	-1	-1	-38

Table D.3 Assessment selection 2: heading & flight-path step response

T	0	1.999	2	3	3.001	40
R E_F	0	0	1	1	1	1

Table D.4 Assessment selection 3: roll and heading at engine failure

airspeed step response						
T	0	1.999	2	3	3.001	40
V_C	0	0	1	1	1	1
UV_C	0	0	1	1	1	1
X_C	0	0	0	1	1	38

wind response						
T	0	1.999	2	3	3.001	40
WXE	0	0	-13	-13	-13	-13

Table D.5 Assessment selection 4: airspeed step & wind response

altitude to airspeed cross coupling						
T	0	1.999	2	3	3.001	40
WV_C	0	0	-30	-30	0	0
Z_C	0	0	0	-30	-30	-30

airspeed to altitude cross coupling						
T	0	1.999	2	3	3.001	40
V_C	0	0	13	13	13	13
UV_C	0	0	13	13	13	13
X_C	0	0	0	13	13	494

Table D.6 Assessment selection 5: airspeed - altitude cross coupling

controller off						
T	0	0.1	0.2	0.3	...	103
CTLR_OFF	1	1	1	1	...	1
WXB	longitudinal turbulence					

controller on						
T	0	0.1	0.2	0.3	...	103
CTLR_OFF	0	0	0	0	...	0
WXB	longitudinal turbulence					

Table D.7 Assessment selection 6: PSD longitudinal turbulence response

controller off						
T	0	0.1	0.2	0.3	...	103
CTLR_OFF	1	1	1	1	...	1
WYB	lateral turbulence					

controller on						
T	0	0.1	0.2	0.3	...	103
CTLR_OFF	0	0	0	0	...	0
WYB	lateral turbulence					

Table D.8 Assessment selection 7: PSD lateral turbulence response

controller off						
T	0	0.1	0.2	0.3	...	103
CTLR_OFF	1	1	1	1	...	1
WZB	vertical turbulence					

controller on						
T	0	0.1	0.2	0.3	...	103
CTLR_OFF	0	0	0	0	...	0
WZB	vertical turbulence					

Table D.9 Assessment selection 8: PSD vertical turbulence response

In order to be able to use the same excitations during flight conditions, other than straight and level flight, the input signals are augmented automatically in those other conditions. (E.g. during descent a decreasing height reference will be accounted for.)

D.5 Plotting process

The results of the assessments are plots, presenting the most important outputs for each assessment. An identification string, automatically printed on the figure, indicates the parameter combinations involved.
The plotting is started by entering:

```
plasm
```

at the Matlab prompt. The script responds by asking for the wanted type of assessment. In order to produce compact results, in a number of cases two types have been combined in one selection. Enter the applicable number to make a selection:

lateral & altitude step response	(1)
heading & flight-path step response	(2)
roll and heading at engine failure	(3)
airspeed step & wind response	(4)
airspeed - altitude cross coupling	(5)
PSD longitudinal turbulence response	(6)
PSD lateral turbulence response	(7)
PSD vertical turbulence response	(8)

Then the parameter combinations for which the plot(s) will be produced have to be specified. The parameter codes available in the data base are shown. This, however, does not imply that <u>all</u> combinations of those codes have been simulated, e.g. when the data base has been created during subsequent partial assessments. It will be checked by the script and warning messages will be issued if necessary.

At each question, any combination of the shown possibilities may be given, in any order or number. Default, by hitting only the Enter-key, all available parameters are specified. The interactive part is finished by a question on separate or multiple plots. By default separate plots are selected.

When separate plots are chosen each parameter combination is dealt with separately. By interactive manipulation it is possible to step back and forth in the plotting data base, by single or by five steps. In this way the designer can acquaint himself with the results and prepare annotations on any behaviour not in accordance with the specifications.
In a number of assessment plots, boundaries have been drawn, indicating the applicable limits specified in section 3.2 for altitudes above 305 m.
When multiple plots were selected, all plots are drawn in one figure. Then the user may annotate any behaviour that does not meet the specifications in an interactive, computer aided way, see Section D.6. After finishing the annotations, the figures are stored as EPS- and PS-files with file names in the format: `nlasm*.eps` and: `nlasm*.ps` where `*` indicates the assessment selection number. Separate plots are not saved automatically. The annotation data base is saved as: `anno*.mat` . For each assessment selection the latest EPS- and PS-file is retained. So, if an earlier version should be kept, the user has to rename it to another name.

When during the plotting process some data appears to be missing, an error message is issued, specifying the missing data. The plotting process is continued with the next plot, but automatic figure saving is inhibited. In that case the script:

```
nlasm
```

has to be started again, and the applicable inputs have to be given. Only the missing data will be calculated.
Some results calculated during assessments 6, 7 and 8 are shown in the command window during separate plotting. During multiple plotting these results are written to an ASCII-file with a name in the format: `nlasm*.asc` where `*` is the assessment selection number.

D.6 Detailed instructions to perform the RCAM Assessment

1. Make sure that you create a new MEX-file corresponding to the latest version of `rcamex.c` in the assessment directory `~/rcam-asm` , revision denoted by:

   ```
   Rev. by D. Moormann (DLR), Aug-13-1996
   ```

   You can create the MEX-file with the command:

   ```
   cmex rcamex.c
   ```

at the operating system prompt, from outside Matlab. You should not copy it from `~/rcam-des` or `~/rcam-eva` , as these directories most likely contain a previous version of `rcamex.c` and the corresponding MEX-file.

2. Make sure that you have the trim data `rcx*$%#.mat` where `*$%#` indicates at least all combinations for:

$$
\begin{aligned}
* &= 1,2 \\
\$ &= 1,2 \\
\% &= 1,2 \\
\# &= 0,1,3,5,6
\end{aligned}
$$

   If not, you may either copy the data from the NLR ftp-site or generate it yourself with the command:

```
trimrcmx
```

   Note that this will take quite some time.

3. Check if you need to incorporate statements in `userasm.m` to initialize your own controller.

4. Check if the integration algorithm uses correct parameters for your controller. The current default uses a maximum integration stepsize of 0.1 s, whereas a typical high-bandwidth controller may need a smaller stepsize, like 0.01 s. The integration algorithm applied in the assessment is `rk45` . Its parameters are specified in `userasm.m` which may be edited by the user, if needed.

5. To perform the assessment, run `nlasm` 8 times subsequently. At the first question:

```
What type of assessment do you want ?
```

   enter `1` up to `8` in the subsequent assessments.

6. The first question may be followed by a question:

```
Do you want to delete the existing plotting data
for this assessment (y) or not [n] ?
```

   This question will only appear when you did the same assessment before.
   Only when you want to delete all your existing data, e.g. when you did implement a new controller, enter `y` , otherwise enter `n` or only hit the Enter-key which implies the default: do not delete.

7. At the next question:

```
Time delay:  nominal [0], minimum (1), maximum (2) >
```

enter `2` in assessments 1 until 5,
enter `0` in assessments 6 until 8.

8. At the next three questions:

```
Aircraft mass:  nominal [0], minimum (1), maximum (2) >

Delta x CoG: mid [0], forward (1), aft (2) >

Delta z CoG: mid [0], lower (1), upper (2) >
```

always enter `12`

9. At the last question:

```
Extra cases:
[0] nominal        (1) R eng fail                 (2) L eng fail
(3) R turn         (4) L turn                     (5) steep descent
(6) max flap speed (7) original design speed 80 m/s    >
```

enter `06531` in assessments 1, 2, 4, 5,
enter `0653` in assessment 3,
enter `0` in assessments 6, 7, 8.

10. Then the simulation process is performed. It will take many hours, depending on the performance of your platform and on the bandwidth of your controller. Any parameter combination dealt with in earlier executions of `nlasm` for the assessment considered, will not be duplicated. All results are stored in a data base, in many MAT-files.

11. To perform the plotting, run `plasm` 8 times subsequently. At the first question:

```
What type of assessment do you want ?
```

enter `1` up to `8` for the subsequent plots.

12. At the next five questions, like in the example shown below:

```
Time delay:  nominal (0), minimum (1), maximum (2)

AVAILABLE CASES : 2    >

Aircraft mass:  nominal (0), minimum (1), maximum (2)

AVAILABLE CASES : 1 2    >

Delta x CoG: mid (0), forward (1), aft (2)
```

```
AVAILABLE CASES : 1 2    >
Delta z CoG: mid (0), lower (1), upper (2)
AVAILABLE CASES : 1 2    >
Extra cases:
(0) nominal          (1) R eng fail                (2) L eng fail
(3) R turn           (4) L turn                    (5) steep descent
(6) max flap speed   (7) original design speed 80 m/s
AVAILABLE CASES : 0 1 3 5 6    >
```

you have to select those cases that are required by the Evaluation Committee (see steps 7, 8 and 9). When just those cases are available you may only hit the Enter-key, to obtain the default: all available cases, When not all required cases are available, you have to go back to step 5. Which cases are available, of course depends on the cases dealt with in the previous execution(s) of `nlasm` .

13. At the last question:

```
Do you want separate plots for investigating performance  [s]  or
   multiple plots in one figure  (m)    >
```

only hit the Enter-key, for the default `s` to obtain separate plots.

The first plot of the plotting data base is drawn. In the command window the serial number, the plot identification and the number of available plots is shown, e.g.:

```
Serial number:  1, ident:  td0:m0:x0:z0:ex0, 40 plots available
```

followed by a question on how to proceed through the data base:

```
For next plot step through plotting data base:
[f] forward (default)           (F) five steps forward
(b) backward                    (B) five steps backward
(1) first plot                  (L) last plot
(e) save this plot to EPS-file  (p) save this plot to PS-file
(m) multiple plots              (s) stop plotting    >
```

This question is assumed to be self-explaining. Note, that the input processing is case sensitive. Hitting only the Enter-key results in one step forward through the data base. Always a single plot is shown at a time. When "save this plot ..." is requested, only the depicted plot is saved.

Please, scan through the plotted results and prepare your annotations, if you might notice any deviations from the specified behaviour, requirement boundary crossings, etc.

When some plotting data is missing, an error message is issued. You are requested to repeat `nlasm` and `plasm` for the relevant assessment, from step 5 and on, with input data covering the missing case(s). Only the actually missing cases will be simulated.

14. After preparing your annotations, you may start multiple plotting by entering an `m` . The plotting is repeated for all selected cases, in a multiple plot. When the actual plotting is completed, and no error messages were issued on missing plotting data, the annotations may be entered into the figure by means of computer aided drawing.

    A few messages are issued:

```
You may annotate this figure now
PLEASE PLACE THE CURSOR IN THE FIGURE WINDOW
Only press a key without using the Enter-key
```

    This means that you have to move your mouse-cursor from the Matlab command window, where it normally will be during interactive procedures, into the figure window.
    When answering a question by entering a character, the cursor shall stay in the figure window. (The character will appear in the command window though.) After entering a character, you do not need to use the Enter-key, unless explicitely asked for. It is possible to retrieve the annotation data base from a previous session, if it exists:

```
(f) load existing annotation data from File
(any key) do not load           >
```

    The drawing program starts with its Main menu:

```
=== Main menu:  ===
(r) indicate Region by drawing Rectangle around it
(t) write Text string
(l) draw connecting Line
(s) Stop annotating            >
```

    Suppose some deviations in your plots are caused by the turning flight case. Then you could indicate the affected part of the curves by a rectangle surrounding it, put

a string `ex3` (meaning extended envelope case 3: turn to right) somewhere in the margin and draw a line between rectangle and string to indicate their relationship. You could enhance the appearance of the figure even further by drawing a rectangle around the text string.
All these actions you can realize with the built-in drawing program.
Assume you hit the R-key to draw a rectangle. A menu will appear for rectangle drawing or editing:

```
Draw rectangle:
(c) Create new rectangle
(a) draw rectangle Around text string
(m) Move existing rectangle
(d) Delete existing rectangle
(s) Stop drawing rectangle     >
```

You hit the C-key to create a new rectangle. The next instruction will be:

```
Indicate upper Left corner of rectangle
and press the Left mouse button
```

You move the cursor to a location left above the region you want to indicate an press the left mouse button. In the figure a small corner symbol appears at the intended upper left corner location, and in the command window the next instruction is shown:

```
Indicate lower Right corner of rectangle
and press the Right mouse button
```

You move the cursor to a location right below the region to be indicated and press the right mouse button. Whether you have a 2- or 3-button mouse does not matter.
A rectangle is drawn and the rectangle drawing menu is shown again.
You may move the rectangle you have just drawn by hitting the M-key, selecting it by placing the cursor over the upper left corner and pressing the left mouse button. Then you enter the new position by pointing at the upper left and lower right locations and pressing the left and right mouse buttons respectively, just as creating a rectangle.
The rectangle may be deleted by placing the cursor over the upper left corner and hitting the Y-key.
Stop drawing rectangles by pressing the S-key.

Line drawing is performed in a corresponding way.

Entering text is started by pressing the T-key when in the Main menu. A menu appears:

```
Write text string:
(c) Create new text string
(m) Move existing text string
(d) Delete existing text string
(s) Stop writing text string     >
```

Pressing the C-key produces the instruction:

```
Go to lower left corner of intended string position,
start writing and finish with Enter-key
```

This is the only case in which the Enter-key is used. Type a string and finish with the Enter-key. Allow each character to be drawn before entering a new character. (The Matlab drawing system may be slow.) The string can be moved around and deleted just as rectangles and lines.

A rectangle can be drawn around a string in an automatic fashion, from the rectangle drawing menu. Due to the fact, that Matlab is not exactly WYSIWYG (what-you-see-is-what-you-get) w.r.t. character dimensions, a trick has been applied to give priority to the appearance of the figure when printed, over that on the computer screen.

When the annotation procedure has been completed, return from the drawing program by pressing the S-key from the Main menu.

15. In case no error messages were issued on missing plotting data, the figure is saved in a file in EPS-format, for encapsulating it in a document, and in a file in PS-format, for making hard copies.

16. During the assessments 6, 7 and 8 some calculations were performed, producing rms, maximum and minimum values of a number of variables. The results are shown in the command window during separate plotting. During multiple plotting, the results are written to an ASCII-file with name in the format: `nlasm*.asc` where `*` is the assessment selection number.

D.7 Problems

In case of any problems, please send an *e-mail* to: `schuring@nlr.nl` .

E Evaluation Questionnaire

GARTEUR ROBUST FLIGHT CONTROL DESIGN CHALLENGE - EVALUATION QUESTIONNAIRE

VERSION: 3

DESIGN ENTRY:

EVALUATOR: DATE:

It is recognised that an evaluators background and experience (see questions 5(a) and 5(b)) will determine which questions they can answer easily. The notes given below each question are aimed at clarifying them and helping assessors to make decisions, by providing specific detailed answers for each question.

The first four questions follow the four assessment criteria given in the benchmark definitions, GARTEUR/TP-088-3 (section 4.1) and GARTEUR/TP-088-4 (section 5.1) respectively.

1(a) TUTORIAL VALUE

One of the aims of Action Group on Robust Flight Control FM(AG08) was that of "transferring the knowledge and experience gained through the Action Group into industry". The design entries support this aim and are here being assessed in terms of their value in meeting this aim. It is therefore the educational content and suitability for publication of the entry that is being evaluated:-

How do you rate the TUTORIAL VALUE of this design entry ? 1 2 3 4 5 ?

1 -> Unacceptable, incomprehensible.
2 -> Unsatisfactory; needs improvement before publication.
3 -> Acceptable; can be published.
4 -> Good, suitable for publication.
5 -> Very good; recommended reading.

COMMENTS:

2. EFFORT NECESSARY FOR APPLICATION

2(a) LEVEL OF UNDERSTANDING

This question is aimed at indicating the level of investment industry will need to make in terms of training of their designers, to be able to use (and if necessary, modify) the method, such that satisfactory results can be achieved. It must be assumed that the method is supported by a user-friendly state-of-the-art toolset.

What LEVEL OF UNDERSTANDING of the methodology is needed to obtain satisfactory results ? 1 2 3 4 5 ?

1 -> Very high, needs detailed knowledge of the principles, the mathematical background and software implementation.
2 -> High, needs detailed knowledge of principles and the mathematical background.
3 -> A lot, needs detailed knowledge of the principles.
4 -> Some understanding of the basic principles.
5 -> Nothing, i.e. 'black box' method.

COMMENTS:

2(b) LEARNING CURVE

Having established the level of understanding required, it is also important to determine whether this can be achieved quickly or whether it will take a long time to acquire the required level of understanding. i.e. the learning curve is:-

How do you rate the LEARNING CURVE associated with using this method, i.e. how easy is the method to grasp ? 1 2 3 4 5 ?

1 -> Very steep; very difficult.
2 -> Steep; difficult.
3 -> Moderate; acceptable effort.
4 -> Gentle; easy.
5 -> Almost flat; very easy.

COMMENTS:

```
2(c) RE-DESIGN
--------------

In practice, it is usual to design flight control systems to be capable of
dealing with a range of operating conditions and for carriage of internal and
external stores. Additionally, new moding might be added to the baseline
system at a later date. At any stage of the design process, an engineer,
other than the original designer, might inherit design responsibility and we
must take into account the ease with which he could take over the design:-

How easy would it be for you to take over the design            1 2 3 4 5 ?
and carry out a RE-DESIGN, with the same method ?

1 -> Very difficult.
2 -> Difficult.
3 -> Normal effort.
4 -> Easy.
5 -> Very easy.

COMMENTS:




2(d) FLIGHT CONTROL LAW STRUCTURES
----------------------------------

This question is looking at the possiblities for using alternative control
strategies which are beyond the current design challenges, by considering the
application of the methodology to the range of flight vehicles that an
organisation might be designing. i.e. it is an assessment of the range of
applicability of the method for flight control.

Does the method support all FLIGHT CONTROL LAW STRUCTURES       1 2 3 4 5 ?
that you might possibly want to design ?

1 -> None.
2 -> Some.
3 -> A lot.
4 -> Most.
5 -> All.

COMMENTS:
```

```
2(e) DESIGN REQUIREMENTS
------------------------

In a similar manner to 2(d), this question is also addressing the potential
of the method in terms of its application beyond the design challenge. i.e.
inclusion of an organisation's flight control design requirements for current
or future project vehicles, in addition to the those used in the design
challenge.

Is it possible to translate all of your DESIGN                    1 2 3 4 5 ?
REQUIREMENTS into the design method syntax ?

1 -> None.
2 -> Some.
3 -> A lot.
4 -> Most.
5 -> All.

COMMENTS:


3. COMPLEXITY OF THE CONTROL SOLUTION
-------------------------------------

3(a) VISIBILITY
---------------

It must be recognised that there are a wide range of specialists and managers
who need to work with the control laws at a later stage of the total design
process. These might be piloted simulation engineers, flight control computer
implementers and testers, or engineers and managers responsible for flight
clearance. The level of visibility of the functionality is important to these
people, to help them to carry out their tasks:-

Do you consider the controller structure presented to             1 2 3 4 5 ?
have good VISIBILITY in terms of its functionality ?

1 -> Unacceptable, no visible correlation between controller structure and
     functionality.
2 -> Unsatisfactory; unclear in some parts.
3 -> Acceptable, controller structure and functionality correlate.
4 -> Favourable level of visibility; easy to follow.
5 -> High level of visibility; easy to understand.

COMMENTS:
```

3(b) COMPLEXITY

The controller algorithms need to be executed in real time in the aircraft's flight control computer. Since this capability is always limited, it is important that control algorithms are efficient and do not lead to real-time processing problems. High order controllers, multi-dimensional look-up tables and complicated nonlinear fuctions all add to this potential problem.

How do you rate the COMPLEXITY of the design, in relation to the design problem complexity ? 1 2 3 4 5 ?

1 -> Unacceptable, too complex.
2 -> Unsatisfactory, very complex.
3 -> Acceptable, complexity of controller structure and design problem is well balanced.
4 -> Favourable; design of low complexity.
5 -> Optimal, all requirements can be fulfilled with a very simple design.

COMMENTS:

3(c) IMPLEMENTATION

Execution time was addressed by the previous question, but there are other aspects which could cause implementation difficulties. For example, the control algorithm's numerical integrity, timing requirements and potential for gain scheduling are all important implementation aspects.

How suitable is the design for IMPLEMENTATION in an aircraft's on-board flight control computer ? 1 2 3 4 5 ?

1 -> Unacceptable, completely unsuitable.
2 -> Unsatisfactory, undesirable effort necessary.
3 -> Acceptable, but modifications necessary.
4 -> Favourable, few modifications necessary.
5 -> Highly suitable, straightforward implementation is possible.

COMMENTS:

3(d) QUALIFICATION AND CERTIFICATION

If the aircraft and its flight control system are to be accepted by the customer it must be 'qualified' against its specification. Before it flies, it must be 'certified' against airworthiness criteria. The method and the resulting design specification need to be compatible with existing regulations for aircraft response and handling qualities.

How suitable is the design for compliance with your QUALIFICATION AND CERTIFICATION procedures ? 1 2 3 4 5 ?

1 -> Unacceptable, certification is impossible.
2 -> Unsatisfactory, undesirable modification effort is necessary.
3 -> Acceptable, but modifications necessary.
4 -> Favourable, few modifications necessary.
5 -> Highly suitable, fits without modification.

COMMENTS:

4. BEHAVIOUR OF THE CONTROLLER

Quantitative measures will be available from the assessment software. Here it is intended to capture a qualitative assessment.

4(a) ROBUSTNESS

Do you have any comments regarding the ROBUSTNESS of the design that has been achieved ?

COMMENTS:

4(b) PERFORMANCE

Do you have any comments regarding the PERFORMANCE of the design that has been achieved ?

COMMENTS:

4(c) CONTROL SURFACE ACTIVITY

Do you have any comments ragarding the CONTROL SURFACE ACTIVITY associated with the design ?

COMMENTS:

5. BACKGROUND AND GENERAL COMMENTS OF THE EVALUATOR

5(a) KNOWLEDGE OF THE DESIGN METHOD

This is to determine the assessors experience with the method to establish the assessors 'authority' with respect to answering the questions. This additional information could be helpful for supporting the interpretation of the total evaluation.

How much KNOWLEDGE OF THE DESIGN METHOD did you (as an evaluator) have before reading the design entry report ? 1 2 3 4 5

1. No knowledge.
2. Some knowledge.
3. Theoretical knowledge.
4. Theoretical knowledge and practical experience.
5. As 4; and having used the design method for a similar problem.

COMMENTS:

5(b) LEVELS OF EXPERTISE

Please indicate your LEVELS OF EXPERTISE in the following areas asociated with flight control systems design, implementation, testing and certification:-

Flight mechanics / Aircraft stability	1 2 3 4 5
Flight Control laws design	1 2 3 4 5
Piloted simulation / handling qualities	1 2 3 4 5
FCC implementation and testing	1 2 3 4 5
Flight clearance / certification	1 2 3 4 5
Flight test / post-flight analysis	1 2 3 4 5

1 => Very little knowledge and experience.
2 => Basic knowledge and some experience.
3 => Good knowledge and experience.
4 => High level of knowledge and experience.
5 => Expert.

COMMENTS:

5(c) GENERAL COMMENTS AND OBSERVATIONS

Taking into account the results achieved and any background knowledge that you may have on the approach considered, please note down any significant additional thoughts or observations that you have made, which have not been covered by the above questions (one page maximum please):-

F Aircraft Physical Modelling leading to Automatic Code Generation

Models of aircraft dynamics should be described in a notation close to the aircraft physics. The most natural way of modelling physical systems is as physical objects and phenomena, which are connected according to their physical energy flow interaction. This is different from modelling via signal flows or input-output block diagrams as traditionally used for controller modelling.

Simulink has been agreed upon as common basis for controller modelling and controlled systems simulations for all participants of the Design Challenge. The Simulink aircraft dynamic blocks for both RCAM* and HIRM† aircraft models are generated automatically from a physically set-up generic aircraft model library developed within the object oriented modelling environment *Dymola* [21, 22].

F.1 Object modelling

An aircraft consists of a variety of different systems, which represent the interacting disciplines involved in aircraft engineering (e.g. flight mechanics, aerodynamics, propulsion). As displayed in Figure F.1, an aircraft consists of a **body** (fuselage and wing), which is powered by one or more **engine**s and which has **gravity** acting on it. The **aerodynamics** describes the effects of the airflow over the aircraft, which is influenced by the surrounding **atmosphere** and additional **wind**s.

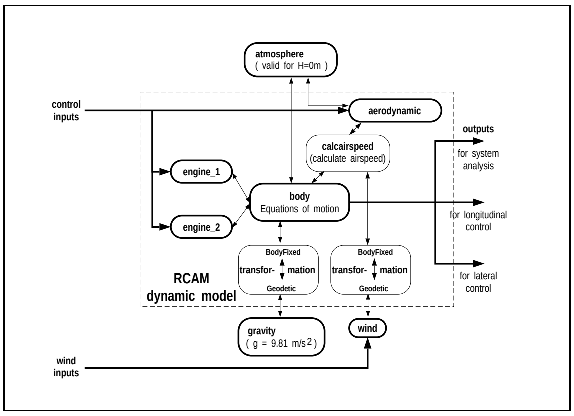

Fig.F.1 Object diagram of RCAM

Each of these phenomena is most conveniently described as one physical object. All objects are connected according to Figure F.1 to represent the interactions within an aircraft.

In order to make the understanding of the objects easy, each component is described in

*RCAM data provided from CERT-ONERA/Aérospatiale (FR)

†HIRM data provided from DRA (UK)

its own coordinate system. Gravity, wind, and atmosphere are conveniently described in an earth related coordinate system, aerodynamics in an wind coordinate system, and engines in a system which is related to the body-fixed coordinate system. Hence coordinate transformations and an object to describe the relationship between velocity, wind, and airspeed are needed in between all of these subsystems when they are connected. Therefore, in addition to the basic aircraft components, coordinate transformations are also detailed and handled as objects in the aircraft library, Figure F.2.

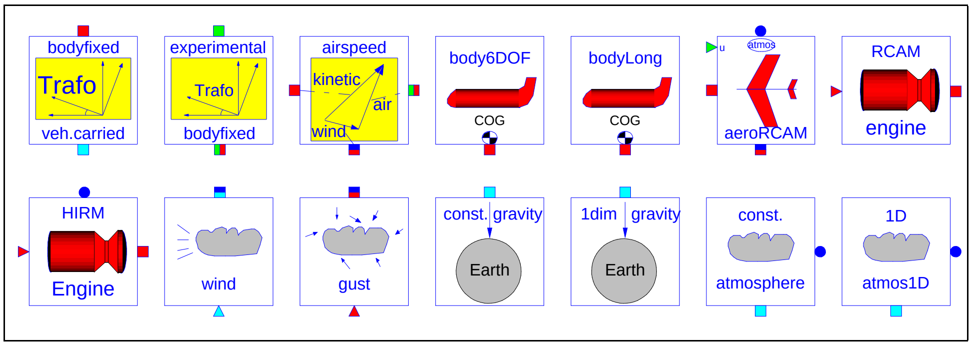

Fig.F.2 Aircraft model library

In the physical aircraft library different representations of one component can be found. There is a class Body with six degrees of freedom (Body6DOF) and a class with three degrees of freedom (BodyLong), which can be used to generate a nonlinear simulation model for the longitudinal axis only. There are also engine, atmospheric and gravity models of different complexity.

In a graphical view (Figure F.1), the interconnection structure of an aircraft can be most easily understood. If a more complex gravity model acts on the aircraft, this object can simply be taken from the aircraft library to replace the simple gravity object. In the same way one or more engines can be added or removed from the aircraft or can be modified. This is the most transparent user layer with no need to think about the structure of any specific simulation code.

The objects which form the physical model contain equations (and not assignments as common in programming or simulation languages). This makes the understanding and the reuse much easier than looking at low level code, whose purpose is to be understood by a computer. Once the objects are available in computer readable form the object equations can be sorted automatically by a symbolic equation handler. This is a main feature of Dymola.

Objects, formulated in that way do necessarily have to represent causalities. This allows one object to fulfill different tasks. For example, the object which does the transformations between the body-fixed and the aerodynamic coordinate system, is used for the transformation of the velocities from the body-fixed system to the aerodynamic system, as they are required within the aerodynamics. The same object is used for the transformation of the forces and moments from the aerodynamic to the body-fixed system. When connecting components as objects, only the relation between them is defined and not the order, in which those equations are finally solved.

In *Dymola*, graphical syntax components are coupled by drawing a line between the defined 'coupling' points of the objects, which are called 'cuts'. These couplings represent either energy or signal flow. For example, the cut *bsystem* (body-fixed system) has the following structure:

$$\text{terminal } {}^{v}T^{b}[3,3], r[3], v_b[3], w_b[3], a_b[3], z_b[3], F_b[3], M_b[3]$$
$$\text{cut bsystem } ({}^{v}T^{b}, r, v_b, w_b, a_b, z_b / F_b, M_b)$$

The matrix ${}^{v}T^{b}$ defines the orientation of the body-fixed system with respect to the vehicle-carried (Earth) system, the vector r is the aircraft's inertial position in the vehicle-carried frame; the vectors v_b and a_b are the velocity and acceleration in the body-fixed frame and the vectors F_b and M_b are the forces and moments, also formulated in the body-fixed frame. In the same way there are cuts defined for the vehicle-carried system (*vsystem*) and for the aerodynamic system (*asystem*)

This cut structure represents physical connections. When objects are connected, *Dymola* adds equations for the cut variables. All quantities of the cut before the slash operator (Across variables) are set equal when connected, as it is reasonable for positions, velocities and accelerations, quantities after the slash operator (Through variables) are summed up to zero, as it is reasonable for forces and moments. This principle is used for connecting engines to the aircraft body for example. The engines have the same position, airspeed, and accelerations than the aircrafts body, their forces and moments sum up with all the other forces and moments acting on the aircraft. Because of that formulation it is easy to add more engines to the aircraft just by adding another engine object to Figure F.1 and connecting it to the aircraft's body.

This object-oriented equation-based form of describing physical systems helps to understand the physical system and enables the user to modify the model most conveniently.

F.2 Hierarchical object structure

An important aspect in object oriented modelling of physical systems is the encapsulation of objects. The internal implementation of details, e.g. of the aerodynamics, are not visible, when viewing the RCAM object model as depicted in Figure F.1. By encapsulation, the implementation of an object can be changed without affecting the functionality of the whole model.

Figure F.3 demonstrates, how the RCAM model is structured. Here only the aerodynamics model is extracted. In the same way details of the engine, gravity, wind, and atmospheric models can be displayed.

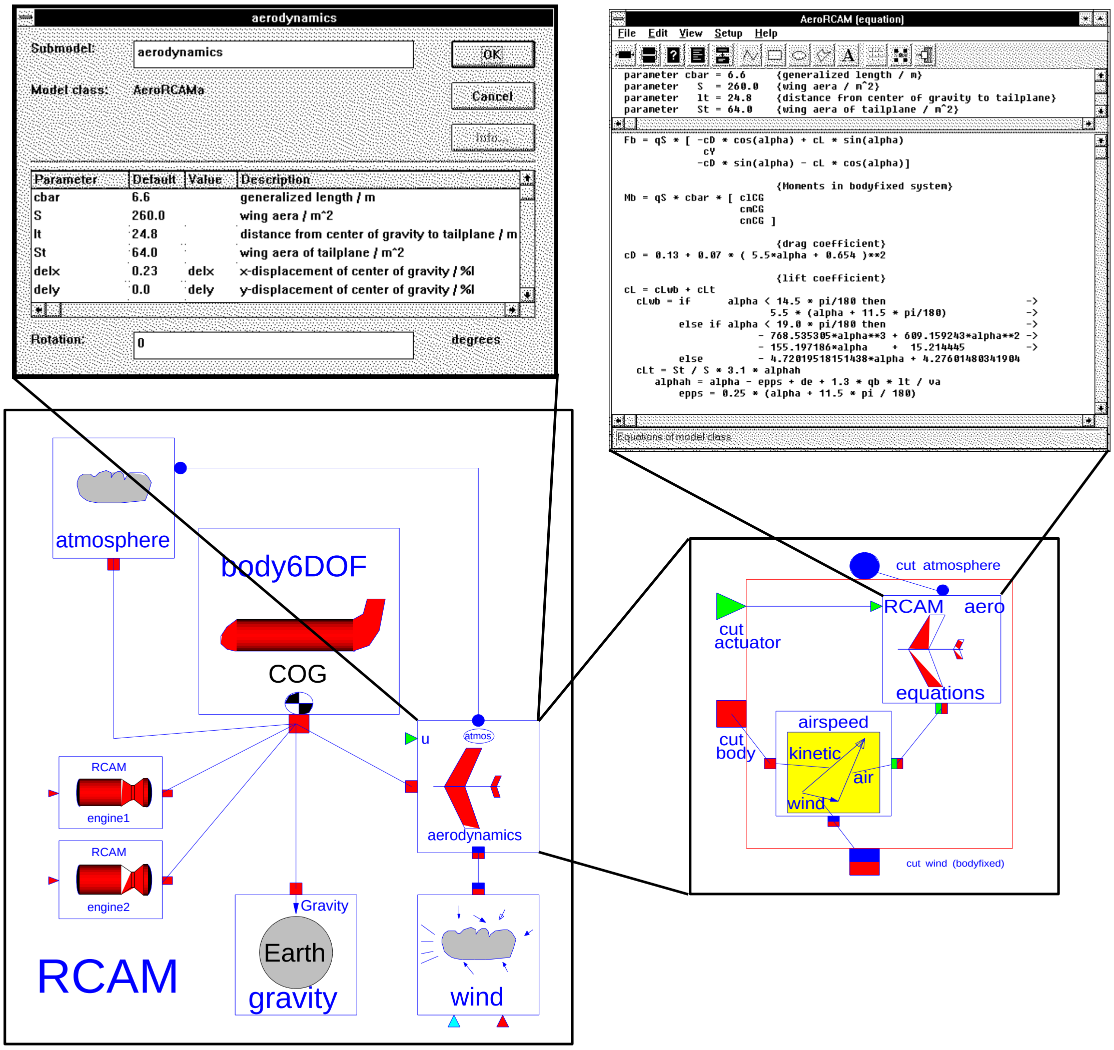

Fig.F.3 Structure of aircraft physical model

Extracting the **aerodynamics** results in the aerodynamics submodel, which consists of the aerodynamic equations **aero equations** and the object **airspeed**, which describes the relationship between the inertial movement, the wind, and the movement relative to the air.

Using the graphical interface, 'double clicking' on **aerodynamics** displays the parameter window of this object. This window allows the parameters to be modified. In the same way, all of the other objects (body, engines) can be instantiated with their parameters.

The objects of RCAM model will be detailed in the following sections. Boxes in the following subsections contain Dymola code in the form of equations.

F.2.1 Object: Body6DOF

The object **Body6DOF** describes the differential equations of motion for a rigid body with six degrees of freedom. The equations of this object are from Section 2.3.1.
It contains the following model parameters, which are the RCAM mass and the moments and products of inertia as detailed in Section 2.3.1:

```
parameter  mass = 120000.0 { mass / kg },
             Ix = 40.07 * mass      { x-moment of inertia / kg*m^2 },
            Ixy =  0                { xy-product of inertia / kg*m^2 },
            Ixz = -2.09323 * mass   { xy-product of inertia / kg*m^2 },
             Iy = 64 * mass         { y-moment of inertia / kg*m^2 },
            Iyz =  0                { yz-product of inertia / kg*m^2 },
             Iz = 99.92 * mass      { z-moment of inertia / kg*m^2 }
```

Translational motion
The equations for the translational movement can be given by the force equation:

```
{ translational equations of motion  }
     Fb = mass * ( ab + cross(wb,vb) )

        ab = der(vb)
        vv = der(r)
```

where Fb is the sum of forces due to the engine, the aerodynamics and gravity, formulated in the body-fixed coordinates, vb is the inertial velocity and wb is the rotation rate in the same coordinate system. The acceleration ab is the time derivative of velocity vb; the velocity vv is the time derivative of the position vector r expressed in the vehicle-carried (earth) frame
Furthermore some aircraft specific quantities are defined:

```
{  height }
   h = -r(3)

{  load factor, as it is used for RCAM: }
{  sum of all forces acting at the aircraft (Fb) minus}
{  the gravity force (solved in body-fixed system) }
{  [n is normalized with the gravity force]}
   n = 1/(mass*9.81) * [ Fb(1) + sin(theta)*mass*9.81              ;
                         Fb(2) - sin(phi)*cos(theta)*mass*9.81     ;
                         Fb(3) - cos(phi)*cos(theta)*mass*9.81 + 1 ]

{  flight path angle }
   gamma  = atan2(-vv(3), sqrt(vv(1:2)'vv(1:2)))

{  flight path heading angle }
   xhi = atan2(vv(2), vv(1))

{  ground speed / (m/s) }
   vground = sqrt ( vv' * vv )
```

Rotational motion

The equations of motion for the rotational movement of a rigid body in the body-fixed axis system form the moment equation for Mb:

```
{ rotational equations of motion  }
Mb = I * zb + cross(wb,(I*wb))

                  { Tensor of inertia
   I = [ Ix , -Ixy,  Ixz ;
        -Ixy,  Iy , -Iyz ;
         Ixz, -Iyz,  Iz  ]

                  { rotational acceleration z = time derivative of
                  { inertial rotation w / body-fixed frame }
   zb = der(wb)

                  {  Time derivative of Euler Angles Phi }
   der (Phi) = MPhid * wb
       Phi = [phi, theta, psi]
       MPhid = [1, sin(phi)*tan(theta), cos(phi)*tan(theta);
                0,       cos(phi)     ,     -sin(phi)      ;
                0, sin(phi)/cos(theta), cos(phi)/cos(theta)]
```

where Mb is the sum of moments about the centre of gravity due to the engine and aerodynamics, wb is the inertial rotational velocity in body-fixed coordinates, and zb is the inertial rotational acceleration in the body-axis system. Using the standard notation [19, 23], the equations listed above are obtained.
The translational differential equations of motion for an aircraft body are typically expressed partly in body axes (dynamics) and partly in the vehicle carried (Earth) frame (kinematics). The same velocity is given in both coordinate systems as body-fixed vb and vehicle-carried vv. Therefore the object TrafoBV of Figure F.2, which includes **all** transformation equations between these coordinate systems, is inherited to the object body.

F.2.2 Object: TrafoBV (Coordinate transformation: body-fixed ⇔ vehicle-carried)

The rotations between the body-fixed and the vehicle-carried coordinate systems are defined in the object TrafoBV. The equations are from Section 2.3.2.
This object contains two cuts. The **cut bsystem** contains all of the kinematic quantities of the body-fixed system. bTv is a [3,3]-transformation matrix between the body-fixed and the vehicle-carried coordinate systems. The cut also contains the inertial position vector r, the translational and rotational velocity, vb and wb respectively, and the translational and rotational acceleration vector, ab and zb respectively (all given in the bodyfixed coordinate system), which are set equal when connected. Behind the slash, the cut contains the body-fixed forces and moments, Fb and Mb respectively, which are summed up when connected.
The cut for the vehicle-carried system **cut vsystem** is defined in the same way.
In *Dymola*-syntax the transformation equations are given as:

```
{ cut of the body-fixed reference frame }
cut bsystem ( bTv, r, vb, wb, ab, zb / Fb, Mb )
{ cut of the vehicle-carried reference frame }
cut vsystem ( vTv, r, vv, wv, av, zv / Fv, Mv )

{ Transformation matrix between body-fixed and vehicle-carried frame
   according to DIN 9300  }
     bTv = [ cos(theta)*cos(psi),
                 cos(theta)*sin(psi),
                    -sin(theta);
             sin(phi)*sin(theta)*cos(psi) - cos(phi)*sin(psi),
                 sin(phi)*sin(theta)*sin(psi) + cos(phi)*cos(psi),
                     sin(phi)*cos(theta);
             cos(phi)*sin(theta)*cos(psi) + sin(phi)*sin(psi),
                 cos(phi)*sin(theta)*sin(psi) - sin(phi)*cos(psi),
                     cos(phi)*cos(theta)                   ]

{ Transformation equations }
     vb = bTv * vv     { velocity vector }
     wb = bTv * wv     { rate vector }
     ab = bTv * av     { acceleration vector }
     zb = bTv * zv     { rotational acceleration vector }
     Fb = bTv * Fv     { force vector }
     Mb = bTv * Mv     { moment vector }
```

where bTv is the transformation matrix between body-fixed and vehicle-carried coordinate system. It transforms all translational and rotational quantities between these two reference frames. Since this object contains equations (and not assignments), it can be used to transform quantities from vehicle-carried to body-fixed coordinates and also to transform from body-fixed to vehicle-carried coordinates. In the latter case, the transformation matrix is inverted automatically.

F.2.3 Object: Aerodynamics

The object **Aerodynamics** describes the RCAM aerodynamics. As depicted in Figure F.3, it consists of an object which defines the relation between inertial speed, airspeed and wind (object Airspeed) and the aerodynamic equations (object Aero_equations).

F.2.3.1 Object: Airspeed (calculation of airspeed)

The calculation of airspeed in *Dymola* -syntax is given by the following equation. The equation for this object is from Section 2.3.3.

```
vab = vb - vwb
```

The airspeed vab, solved in body-fixed coordinates, contains the x-, y-, and z-components of the aircraft's velocity relative to the surrounding air. The airspeed is the difference between the inertial velocity *vb* and the wind *vwb*, which are both also formulated in body-fixed coordinates.

vwb is the sum of translational **wind** and **gust**, as depicted in Figure F.4. Since the **wind** is described most naturally in vehicle-carried (earth) frame it is transformed into to body-fixed system by the corresponding coordinate transformation. No transformation is required for the **gust**, because it is formulated in body-fixed coordinates.

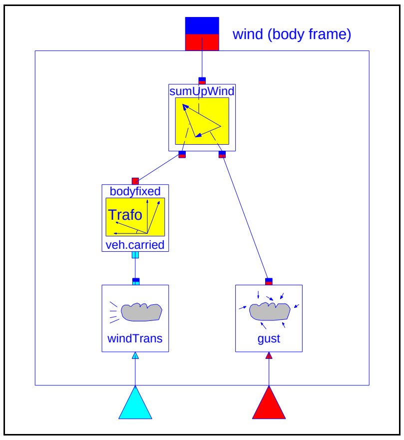

Fig.F.4 RCAM Wind Model

F.2.3.2 Object: Aero_Equations

The aerodynamics contain the following model parameters, which are detailed in Table 2.4. The parameters *cmerr*, *clerr*, and *cnerr* are uncertainty parameters with the bounds given in Table 2.5.

```
parameter cbar = 6.6    { generalized length / m }
             S = 260.0  { wing area / m^2 }
            lt = 24.8   { distance from centre of gravity to tailplane }
            St = 64.0   { wing area of tailplane / m^2 }
          delx = 0.23   { x-displacement of centre of gravity / %cbar }
          dely = 0.0    { y-displacement of centre of gravity / %cbar }
          delz = 0.0    { z-displacement of centre of gravity / %cbar }
```

The equations in Aero_Equations are from Section 2.3.4.

Aerodynamic Forces

The aerodynamic forces are determined by means of aerodynamic coefficients for drag, side force, and lift (*cD*, *cY*, *cL*), which are functions of the angle of attack *alpha*, the sideslip angle *beta*, and the control surface deflections (*da*, *de*, *dr*).

```
{ Aerodynamic force coefficients }
                        { drag coefficient  }
cD = 0.13 + 0.07 * ( 5.5*alpha + 0.654 )**2

                        { lift coefficient  }
cL = cLwb + cLt
  cLwb = if      alpha < 14.5 * pi/180 then                      ->
                        5.5 * (alpha + 11.5 * pi/180)            ->
         else if alpha < 19.0 * pi/180 then                      ->
                      - 768.535305*alpha**3 + 609.159243*alpha**2 ->
                      - 155.197186*alpha    +  15.214445         ->
         else         - 4.72019518151438*alpha + 4.27601480341904
  cLt = St / S * 3.1 * alphah
     alphah = alpha - epps + de + 1.3 * qb * lt / Va
         epps = 0.25 * (alpha + 11.5 * pi / 180)

                        { side force coefficient  }
cY = -1.6 * beta + 0.24 * dr
```

The aerodynamic forces are given in body-fixed coordinates Fb as a function of aerodynamic factors qS, aerodynamic coefficients cD, cY, cL, the airspeed Va, and the angle of attack $alpha$.

```
{ Aerodynamic Forces }
                    { Forces in body-fixed system }
  Fb = qS * [ -cD * cos(alpha) + cL * sin(alpha)
               cY
              -cD * sin(alpha) - cL * cos(alpha)]

                    { dynamic pressure times wing area }
  qS = rho / 2.0 * Va * Va * S

                    { airspeed }
  Va = sqrt (vab'*vab)
  alpha = atan2(vab(3), vab(1))
  sin(beta)  = vab(2) / Va
```

Aerodynamic moments

The moments due to the aircraft aerodynamics are determined by means of the aerodynamic coefficients (cl, cm, cn), which are assumed to act about the aerodynamic centre of the aircraft and are given by the following equations:

```
{ Aerodynamic moment coefficients about aerodynamic centre }
[cl;cm;cn] = clmn0 + clmnRot*[pb;qb;rb] + clmnAct*[da;de;dr]

 clmn0   = [         -1.4 * beta                      ;
             -0.59 - 3.1*St*lt/S/cbar*(alpha - epps)  ;
               (1 - alpha/15.0*180/pi) * beta         ]

 clmnRot = cbar/Va * [ -11 ,          0              , 5    ;
                        0  ,-4.03*St*lt*lt/S/cbar/cbar, 0    ;
                        1.7,          0              ,-11.5 ]

 clmnAct =  [ -0.6,        0        , 0.22 ;
               0  ,-3.1*St*lt/S/cbar, 0    ;
               0  ,        0        ,-0.63 ]
```

The moment coefficients about the centre of gravity ($clCG$, $cmCG$, $cnCG$) are calculated from these aerodynamic centre-based coefficients using the following equations:

```
{ Aerodynamic moment coefficients about centre of gravity }
[clCG;cmCG;cnCG] = [cl;cm;cn] + cross(delta,Fb/qS)

    delta = [delx - 0.12 ;
                dely     ;
                delz     ]
```

The aerodynamic moments Mb are given in body-fixed coordinates as a function of the dynamic pressure times wing area factor qS, the aerodynamic coefficients $clCG$, $cmCG$, $cnCG$, and the generalized length $cbar$.

```
{ Aerodynamic moments }
  Mb = qS * cbar * [ clCG
                     cmCG
                     cnCG ]
```

F.2.4 Object: RCAM_Engine

The RCAM_Engine contains the following model parameters, which are detailed in Table 2.4. Additionally the parameters $delx$, $dely$, and $delz$ are uncertain parameters with bounds according to Table 2.5.

```
parameter xcge        { engine x-pos with respect to centre of gravity / m }
          ycge        { engine y-pos with respect to centre of gravity / m }
          zcge        { engine z-pos with respect to centre of gravity / m }
          delx        { centre of gravity variation in body x / %cbar }
          dely        { centre of gravity variation in body y / %cbar }
          delz        { centre of gravity variation in body z / %cbar }
    nominalMass  = 120000.0 { thrust defining mass / kg }
              xi = 0  { thrust longitudinal angle / rad }
         epsilon = 0  { thrust lateral angle / rad }
```

The thrust F is normalized by the gravity force acting at the RCAM model, which is defined by the product of the nominal $NominalMass = 120\,000$ kg and the gravity acceleration g. For zero thrust angles xi and $epsilon$ the x-component of force Fb is equal to the thrust.

Due to the geometric location of the engines ($xcge$, $ycge$, $zcge$) the engine thrust also contributes to the moments Mb acting on the aircraft.

The equations of RCAM_Engine are from Section 2.3.5.

```
{ thrust calculation }
F = throttle * NominalMass * g
g = 9.81

{ transformation of the thrust vector to body-fixed coordinates }
Fb  = bTeng * [F;
               0;
               0]

{ transformation matrix between engine and body fixed coordinates }
  bTeng = [cos(xi)*cos(epsilon), -sin(xi), cos(xi)*sin(epsilon);
           sin(xi)*cos(epsilon),  cos(xi), sin(xi)*sin(epsilon);
                  -sin(epsilon),    0    ,         cos(epsilon)]

{ thrust moments due to locations of engines and centre of
   gravity variations }
Mb  = cross(cge,Fb)

{ engine displacement vector from centre of gravity }
  cge = [delx - xcge ;
         ycge - dely ;
         delz - zcge ]
```

Note that the engine model also contains the parameters xi and $epsilon$, which define a longitudinal and lateral thrust angle, respectively.

Since the RCAM engine is aligned with the x-body axis, these parameters are zero. Since zero parameters are removed in the resulting code they do not influence its efficiency. However, their formal inclusion allows a greater flexibility in using the engine model also for non-zero parameter values as well.

F.2.5 Object: Atmosphere

The atmospheric parameters are considered to be constants, that means they are independent of height and position. The data of the RCAM atmosphere model come from Section 2.3.6.

```
rho = 1.225
p   = 101325.0
T   = 288.15
a   = 340.3
```

with rho as the density of air, p the static pressure, T the absolute temperature, and a as the local speed of sound.

F.2.6 Object: Gravity

Gravity is most conveniently described in an earth related coordinate system. The forces acting on an aircraft are best understood when given in the body-fixed coordinate system. Therefore the coordinate transformation between body-fixed and vehicle-carried coordinate system as given in Section F.2.2 is needed to transform the general gravity model to an aircraft gravity model.

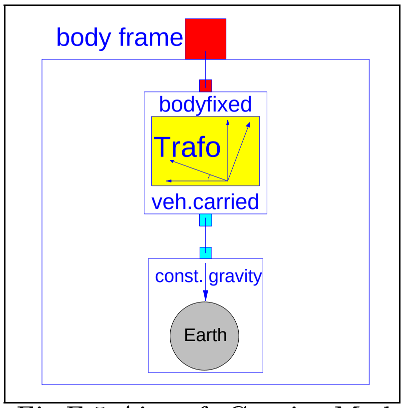

Fig.F.5 Aircraft Gravity Model

Here gravity is considered not to be a function of altitude. The equations of the gravity model are given in the vehicle-carried reference system.
The gravitational constant near the surface of the earth is taken as g = 9.81 m/s^2. For constant gravity, there is only a component in the z-direction of the geodetic system.
The equations for this object are given in Section 2.3.7.

```
parameter g = 9.81

Mv = [0,0,0]
Fv = [0,0,g*mass]
```

F.3 Code generation

From the graphical and textual model description *Dymola* generates efficient code for different simulation environments.
Its symbolic equation handler generates a state space model from the parameter instantiated equations of each object and from the equations derived from the interconnection structure. The equations are sorted and solved according to the specified inputs and outputs. Equations which are formulated in an object but not needed for the specified

configuration are removed automatically. The result is a mathematical model with a minimum number of equations for the specified task.
As a next step, simulation code for different simulation environments (e.g. Simulink, ACSL, ANDECS_DSSIM) is generated automatically. The code for Simulink can be a m-file or a cmex-file. Fortran or C code can be exported in the DSblock neutral simulation-model format [28], to be used in any other simulation run-time environment capable of importing Fortran or C models. This is targeted in particular at the ANDECS design environment for control engineering [24].

F.4 Conclusion and outlook

It is most natural to model physical systems on a physical level in the form of equations. For simulation purposes, simulation code can be generated automatically from physical equations provided that a suitable software tool like Dymola is available. It has been shown that aircraft dynamics code for Simulink is generated for common use in the Robust Control Design Challenge. This is achieved by using a generic *Dymola* aircraft object library.

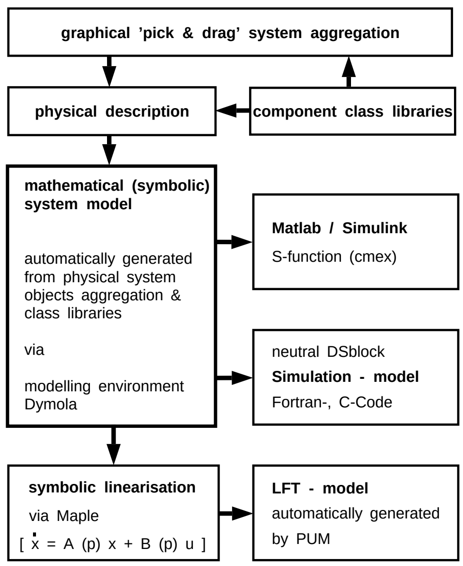

Fig.F.6 From system configuration to symbolic model or simulation model

This approach to automatic code generation has the further advantage that not only can efficient parameterized simulation code be obtained for different simulation and analysis environments but a parameterized symbolic code can be produced as well. This can be used as input for symbolic analysis tools such as PUM [26] (Matlab Toolbox for Parametric Uncertainty Modelling) or PARADISE [30] (PArametric Robustness Analysis and Design Interactive Software Environment).