Proceedings of the EuroGNC 2013, 2nd CEAS Specialist Conference on
Guidance, Navigation & Control, Delft University of Technology, Delft,
The Netherlands, April 10-12, 2013

TECS/THCS-based generic autopilot control laws for aircraft mission simulation

Gertjan Looye

Abstract Mission simulation involves automated simulation of complete aircraft flights or flight segments in order to assess over-all metrics like block fuel, flight times and total emissions. This contribution presents a fully automated mission simulation environment that may be used in flight trajectory as well as aircraft or engine design optimisation loops. To this end, the environment covers the complete process, from integration of a flight dynamics model for the given aircraft configuration to simulation and assessment of all metrics of interest. The aircraft is usually represented by a point-mass, which is guided along 3D trajectories by means of generic autopilot guidance control laws based on the Total Heading and Total Energy Control Systems (THCS and TECS). These control structures elegantly accommodate all speed and path-related autopilot modes, control priorities, performance and protection limits and do not require aircraft-specific gain tuning. This paper discusses the mission simulation environment, the implementation of TECS / THCS herein, as well as example applications on dissimilar aircraft and missions types.

1 Introduction

Mission simulation involves automated simulation of complete aircraft flights or flight segments in order to assess over-all metrics like fuel burn, flight times, and emissions, or to provide trajectory data for subsequent analyses like noise effects on the ground. The methodology is traditionally used in the design of flight procedures, the optimisation of flight trajectories, and over-all design of aircraft and propulsion systems. In case of aircraft design, mission simulation is more and more used even from the preliminary stages, since traditional semi-empirical methods to estimate block fuel and range not necessarily apply to radically new airframe configurations,

Gertjan Looye
Robotics and Mechatronics Center, Institute of System Dynamics and Control, Department of Aircraft Systems Dynamics, e-mail: Gertjan.Looye@DLR.de

propulsion systems or on-board system architectures (e.g. more electric aircraft). In addition, with vastly increasing fuel prices, even marginal improvements are relevant, requiring improved accuracy of numerical analysis approaches early on.

The basic objective of this work has been to develop an accurate mission simulation tool that can be used *both* for aircraft design optimisation purposes *as well as* development of operational procedures. The intended use in aircraft (pre-)design optimisation basically imposes the requirement that the analysis is fully automated, starting from the integration of a simulation model for the current airframe design parameters, simulation of representative design missions, up to computation of relevant performance and environmental metrics to be addressed by the optimisation algorithm.

The main problem in mission simulation is how to make the aircraft model ”fly” a prescribed trajectory or flight procedure in a consistent and repeatable way using engine throttles and control surface deflection commands. The most common approach is to trim the aircraft over a dense grid of flight points along a two-dimensional trajectory (altitude and speed commands), while continuously updating fuel mass, required speed, and configuration. In this work, a more accurate approach is used by performing dynamic simulation of the aircraft in combination with generic autopilot control laws. As compared with trimming, the use of autopilot control laws more accurately reflects the way how aircraft are actually operated in practice and they allow for natural extension of the analysis from two to three dimensional trajectories. Flight dynamics models may be just point mass representations or address the full six degrees of freedom. The first variant is considerably faster, the latter is more accurate but requires the provision of aircraft attitude control laws.

Traditional autopilot control laws need to be tuned to the specific aircraft type, which in the presented environment is not known on beforehand. At this point, the Total Energy and Total Heading Control Systems (TECS, THCS) [10, 11] provide a very elegant solution. One of the basic principles behind these systems is the separation into aircraft independent outer loop (for path and speed control) and aircraft-dependent inner loop control laws (for aircraft attitude and engine control). In case of a point mass approximation, inner loops are obsolete altogether, thus requiring no control law adaptation at all. Besides a generally applicable control law structure, TECS and THCS form a basis for a clean autopilot functional implementation strategy, providing a common core to all autopilot modes, and allowing for straight forward implementation of protection features and control saturation handling [6].

This paper describes how the Total Energy and Total Heading Control Systems and underlying design philosophy have been integrated in the developed mission simulation environment. It is structured as follows. Section 2 briefly discusses the simulation-based analysis process in general terms. In Section 3 the simulation environment and its main components are described. Section 4 describes the implementation of the TECS and THCS control law structures. A number of application examples for different types of aircraft and mission trajectories will be given in Section 5, followed by conclusions in Section 6.

2 The simulation-based analysis process

The mission simulation environment developed in the frame of this work can be used for trajectory optimisation as well as aircraft (or engine) design optimisation. The two applications are illustrated in Fig. 1. For trajectory optimisation, specified trajectory parameters are set by the optimiser, whereas the aircraft model remains unchanged (”Mission Generation” in Fig. 1). In case of aircraft optimisation, aircraft parameters are varied whereas the trajectory is given in the form of fixed design missions or manoeuvres. This application obviously requires the generation of an aircraft flight dynamics model for each newly updated design configuration (”Aircraft Model Integration” in Fig. 1). This is a challenging subject by itself, which will be briefly addressed and referenced in Section 3.2.

With an aircraft model and mission description available, both are integrated respectively loaded into the simulation environment (”Simulation Environment Integration”), allowing the intended simulation analyses to be performed in the next step (”Simulation Analyses”). Based on the analysis results, metrics of interest may be computed, like total fuel burn, gaseous emissions, etc. (”Computation of metrics”). In other cases, trajectory data are passed on to specialised analysis tools, like the PArametric Noise Analysis Module (PANAM) [2].

The process depicted in Fig. 1 is used for aircraft design studies in DLR internal projects like TIVA (Tool Integration for the Virtual Aircraft), VAMP (Virtual Aircraft Multidisciplinary Analysis and Design Processes), and FaUSST (Fortschrittliche aerodynamische UCAV (Unmanned Combat Aerial Vehicle) Stabilitäts- und Steuerungstechnologien – advanced aerodynamic UCAV stability and control technologies). It should be noted that mission simulation is just one of a considerably

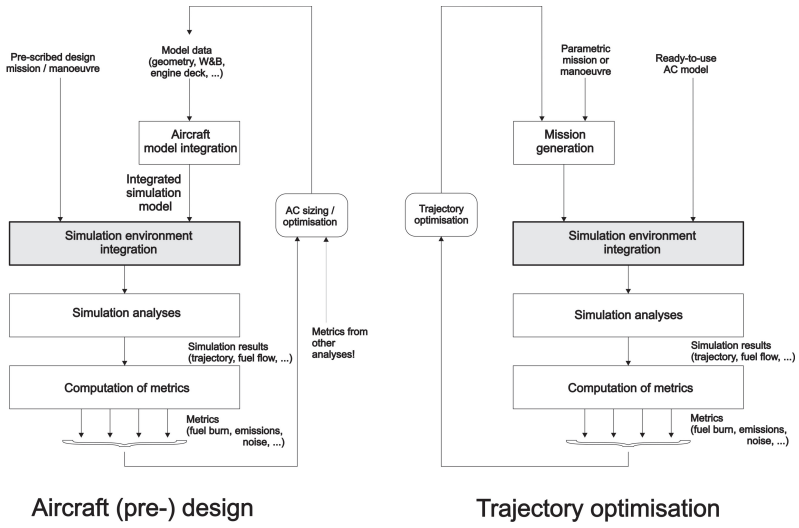

Fig. 1 Mission simulation analysis process

larger number of analyses performed. For trajectory optimisation, the process is for example used in the work described in [20].

In the frame of these projects, all data exchange is performed via an XML-based format called CPACS (Common Parametric Aircraft Configuration Schema) [4]. The whole process obtains its input data and stores its output data in a CPACS-based XML-file. The mission to be flown is specified as well and may be described parametrically for trajectory optimisation purposes. The mission description very closely resembles the one described in [1].

The mission simulation environment is able to handle complete flights (including taxiing, take-off roll, climb, cruise, descend, final approach, go-around, landing, and landing roll-out), or individual flight phases only. For aircraft design purposes, usually so-called design missions are specified that are typical for the intended operations. As an example, Fig. 3 depicts a design mission for an Unmanned Combat Aerial Vehicle (UCAV) as used in the DLR projects UCAV-2010 and FaUSST (see Fig. 2, left). This is a full mission, starting from take-off, climb, cruise, descent, loiter, return to base and landing with the intention to compare fuel burn between different design configurations. Another application is optimisation of final approach paths in order to reduce noise impact on the ground (computed using dedicated external tools, like PANAM [2]). Fig 4 depicts a final approach path, involving arrival at an altitude of around 7500ft, followed by a spiralling descent onto a -3 deg glide path, followed by landing. This approach is called a Helical Noise Abatement Procedure (HeNAP) and, by staying at high altitude as long as possible, intends to concentrate noise of landing aircraft in the direct vicinity of the airport [2]. In this case, comparative studies were made for a passenger aircraft (Fig. 2) between various initial altitudes of arrival (determining the number of full turns to landing), addressing aspects like flight times, fuel burn, and noise footprints on the ground.

UCAV (DLR-F17)

D150 (Airbus A320-like)

Fig. 2 The UCAV (wing span $\approx$ 12m) and D150 configurations (wing span $\approx$ 34m)

3 The simulation environment

The core of the process in Fig. 1 is the integrated simulation environment. Its general structure is depicted in Fig. 5 and consists of the aircraft simulation model, flight

path and speed tracking control laws, guidance algorithms and a trajectory generator. The individual components will be described in the following subsections. The TECS / THCS core will be detailed in Section 4.

3.1 Trajectory generation and guidance algorithms

Trajectories as depicted in Figs. 3 and 4 are defined by means of waypoints and altitude and speed-profiles in between. The task of the Trajectory Generator (Fig. 5) is to generate suitable flight path and speed references from the mission description. This for example involves:

- waypoint navigation: connecting waypoints via great-circle segments, monitor progress along these segments, switch to the next segment such that transitions are smooth, etc. From the great-circle segments, lateral deviations as well as course angles and angular rates are computed depending on the current aircraft position;
- vertical flightpath references: command altitude or specify thrust settings, monitor exit conditions (e.g. maximum climb thrust until a certain flight level (exit condition) is reached), ensure smooth transitions (for an example, see [16]);
- speed references: depending on the flight phase this may be mach number (cruise), calibrated, true, or ground speed. Again, exit conditions are monitored, ensuring smooth transitions.

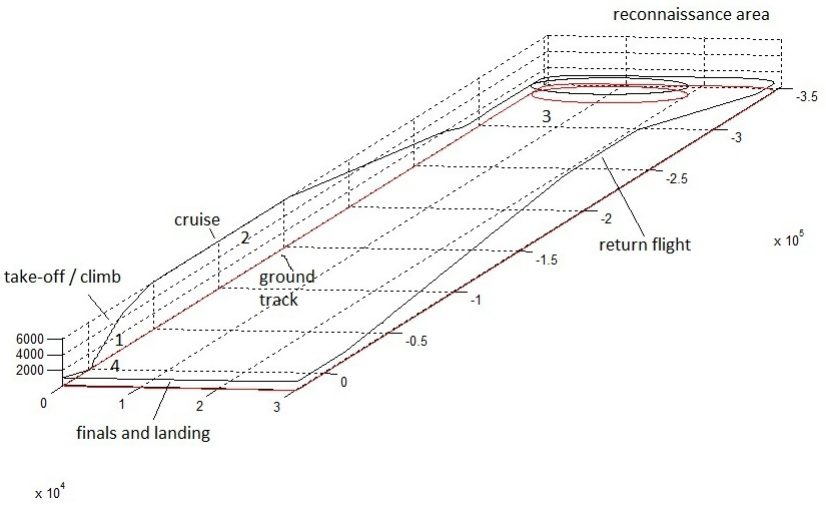

Fig. 3 Example design mission as used for the F17 UCAV depicted in Fig. 2 (left)

- aircraft configuration: specify gear position and high-lift configuration, either commanded from the database or as a function of flight condition by means of exit conditions.

The simulation environment allows missions to be specified fully parametrically, which is a prerequisite for trajectory optimisation applications. Trajectory generation for the helical approach (Fig. 4) is described in detail in [16].

Based on references provided by the trajectory generator, the guidance algorithms in Fig. 5 generate flight path and track angular deviations as well as speed references for the TECS / THCS-based path and speed tracking controllers. The algorithms include a large set of standard autopilot modes that are triggered by the trajectory generator or by internal protection logic (e.g. minimum speed, speed or flight path angle priority in case of thrust saturation, etc.). Especially in aircraft design optimisation, protection modes are triggered frequently as intermediate airframe configurations may have insufficient performance characteristics to fly a given

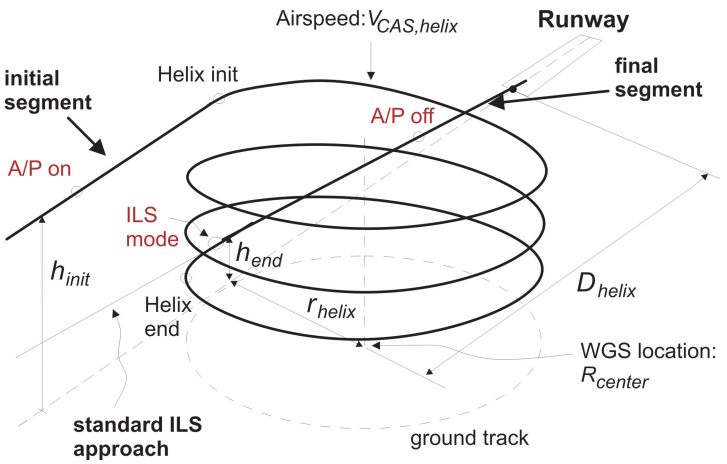

Fig. 4 Helical noise abatement procedure

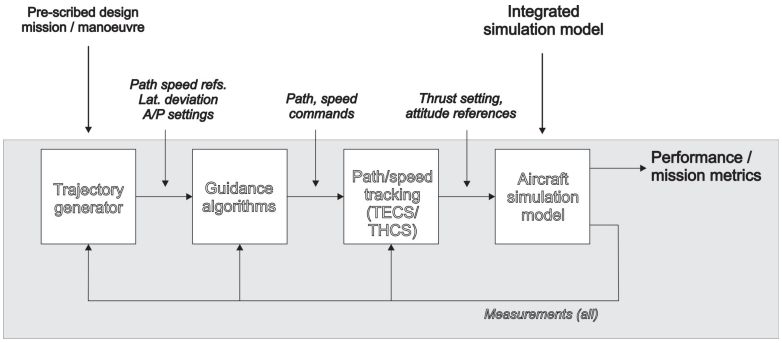

Fig. 5 Aircraft mission simulation structure

design mission segment, or run out of fuel before the mission is completed. In case of trajectory optimisation it may occur that flight path parameters are set beyond performance limits of the given aircraft.

Since mission simulations may involve complete flights, the guidance algorithms also include basic modes that enable automatic take-off, landing, landing roll-out, and taxiing. These will not be elaborated here.

3.2 Aircraft simulation model

The aircraft flight dynamics model obviously is the core of the simulation environment. In case of trajectory optimisation, this model is fixed and may be provided from an external source. In case of an aircraft design loop, the model must be automatically adapted to changing aircraft (geometry) parameters during each iteration. This requires a dedicated modelling process by itself, which will not be discussed here. The interested reader is referred to [7, 8, 15]. For computation of engine parameters like fuel flow and emission rates as a function of throttle setting, an engine deck is integrated. Engine decks for various types of engines are for example obtained from the TWdat tool [].

The basic structure of the aircraft model has been implemented in Modelica [19] using a dedicated flight dynamics library [15], see Fig. 6. Components like aerodynamics, engines, landing gears, sensors, actuation systems, weight and balance, and the airframe (equations of motion, kinematics) are objects interconnected by means of physically based connectors. Model data is obtained from external databases, in this case from an XML-based CPACS file (middle block, see Section 2). In case of aircraft design optimisation, the content of the aerodynamics and weight and balance blocks will change as a function of design parameters.

In order to make the aircraft model fly the reference flight path and speed trajectories, the simulation environment has two options. The first is a natural extension of the traditional trimming approach, namely inverse simulation. This technique involves inverting the model equations, such that these path and speed references are inputs and required control inputs, thrust settings etc. are outputs. Main advantage of the inverse model simulation approach is that it is computationally very fast (see Refs.[20],[21],[9] for more details) and does not require path and speed tracking control laws at all. Main disadvantage is that different model inversions are required in case flight path / speed trajectories are to be tracked, or either of those is tracked at preset thrust levels (e.g. at take-off and climb out).

The second approach described in this contribution is the use of speed and path tracking control laws as depicted in Fig. 5. The implemented model initially has six degrees of freedom (6DOF). In order to interface this model with the control laws, it must be possible to directly control thrust as well as the aerodynamics attitude variables angle of attack (α) to control lift, bank (μ) or roll angle (Φ) to tilt the lift vector for lateral manoeuvring, and angle of side slip (β) or yaw rate ($\dot{\Psi}$) to control

side force. Again, the simulation environment offers two approaches, see also Fig. 7:

1. by reducing the full model to a point-mass approximation with translational degrees of freedom only. The afore mentioned variables are directly set as inputs;
2. by augmenting the 6DOF model with automatically generated dynamic inversion-based stability and command augmentation control laws that track reference commands for the afore mentioned variables.

Since metrics derived from mission simulation usually address global or cumulative aspects of the simulated flight (e.g. total fuel use), relatively fast aircraft attitude dynamics may usually be neglected (afore mentioned point mass approximation). This considerably reduces simulation times due to lower computational effort the possibility of increasing integration time steps. The attitude controlled 6DOF version has been realised anticipating future more dynamic design manoeuvres.

The force equations of motion are expressed in flight path co-ordinates:

$$m \begin{bmatrix} \dot{V} \\ \dot{\chi} V \cos\gamma \\ -\dot{\gamma} V \end{bmatrix} = R_{ke} \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} + R_{kb} \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + R_{kg} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{1}$$

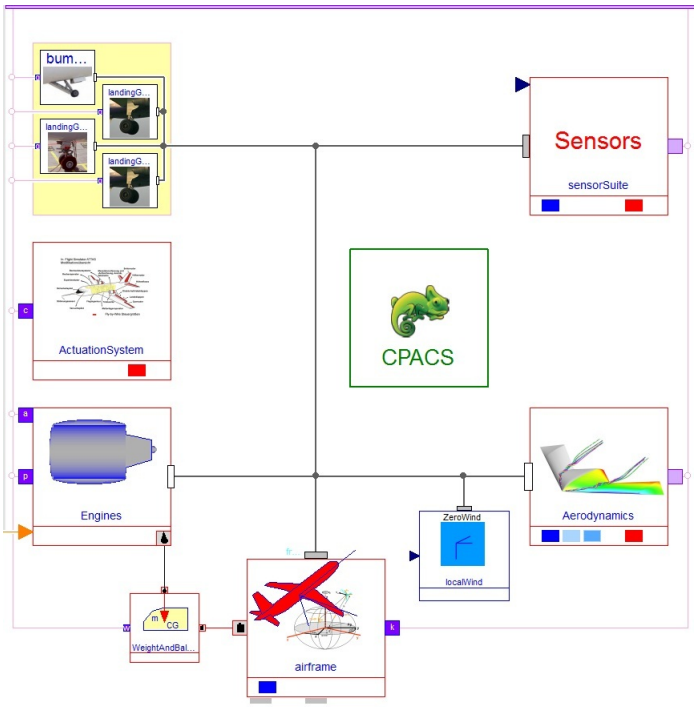

Fig. 6 Aircraft model structure in Modelica

where V is the inertial velocity, γ the vertical flight path angle, χ the flight path track angle, m the aircraft mass, and g the gravitational acceleration. The aerodynamic forces are drag (D) side force (Y) and lift (L), which in this case are expressed in experimental co-ordinates. The thrust components in aircraft body axes directions are T_x, T_y, and T_z respectively. Finally, direction cosine matrices R_{ke}, R_{kb}, R_{kg} transform vectors from respectively experimental, body, and local geodetic co-ordinates into flight path co-ordinates. The equations and reference systems are for example described in [5].

Although the equations have been implemented in Modelica in this way, a simplified formulation will be used for further reference in this paper. With the assumptions $T = T_X$, $T_y = 0$, $T_z = 0$, $\cos\alpha \approx 1$, $\cos\beta \approx 1$, $\Phi \approx \mu_k$, no wind, and neglecting thrust forces acting perpendicularly on the flight path, the equations reduce to:

$$m(\dot{V} + g\sin\gamma) = T - D \tag{2}$$

$$m(V\dot{\gamma} + g\cos\gamma) = L\cos\Phi - Y\sin\Phi \tag{3}$$

$$m\dot{\chi}V\cos\gamma = L\sin\Phi + Y\cos\Phi \tag{4}$$

Eqn. 2 can be re-written as:

$$\frac{\dot{V}}{g} + \sin\gamma = \frac{T - D}{W} \tag{5}$$

with $W = mg$ is the aircraft weight. The right hand term is called *specific excess power*, which is effectively used to accelerate or to climb.

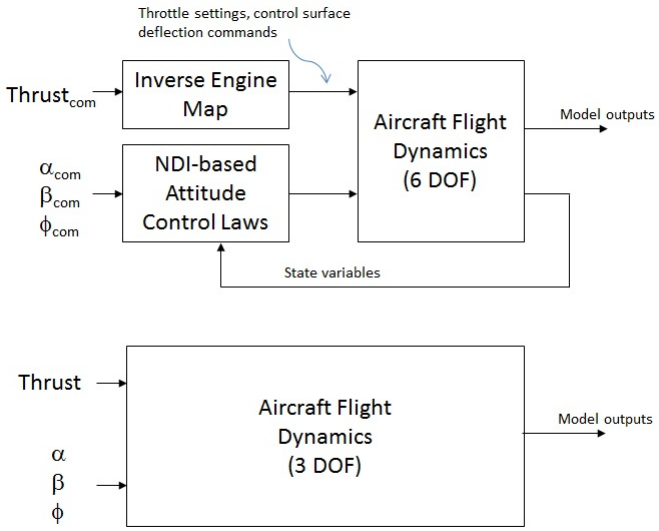

Fig. 7 Aircraft attitude control variants

In Eqn. 3 the effect of angle of attack change $\delta\alpha$ on lift is given by:

$$L = L_0 + \delta L = L_0 + \bar{q}SC_{L_\alpha}\delta\alpha \tag{6}$$

where L_0 is the initial lift force for the undisturbed angle of attack, $\bar{q}$ is the dynamic pressure, S is the aircraft wing area, and C_{L_α} the dimensionless lift gradient. The latter may depend on the flight condition. Starting from a trimmed condition in straight and level flight (i.e. $L_0 = W$), the equation may be re-written as follows:

$$m(V\dot{\gamma} + g\cos\gamma) = (W + \bar{q}SC_{L_\alpha}\delta\alpha)\cos\Phi - Y\sin\Phi \tag{7}$$

When neglecting rotational dynamics around the aircraft body axes, it becomes necessary to specify the attitude be means of external inputs. These may be Euler angles, but a more suitable choice is the triplet aerodynamic angle of attack α to directly control lift, the roll angle Φ to tilt the lift vector for lateral manoeuvring, and either side slip angle β to control side force directly, or yaw rate $\dot{\Psi}$ to co-ordinate turning flight (to be elaborated in the following section).

A major advantage of implementing the aircraft model in Modelica is that point-mass or inverse model equations (for inverse simulation or attitude control laws) can be derived fully automatically from one and the same model implementation [18].

Flight path angle and speed responses for the two example aircraft to a 10% throttle and 1deg angle of attack step input are depicted in Fig. 8. The initial speeds (100 m/s) and altitudes (1000 m) are equal. Obviously, the phugoid frequencies are similar, as the value is approximately proportional to airspeed [5]. Differences arise due to different engine, lift-drag polar characteristics, and weight, see eqns. (2) and (3). Side slip and track angle responses to roll and heading rate step input commands are shown in Fig. 9. These differ considerably between the configurations. This is caused by the absence of a fuselage and vertical tail in case of the F17 configuration. The simplified aerodynamic modelling approach results in zero side force due to side slip angle. As a consequence, in case the aircraft rolls, the pitch attitude angle is partly changed into a side slip angle (initial step). Then the aircraft starts to drift side wards, resulting in an ever increasing side slip angle. In case of a heading rate step, for the F17 UCAV the growing side slip angle does not result in side forces, so that the track angle remains zero. For the D150 configuration, the aircraft performs a so-called flat turn. In case of 6DOF simulations, the F17 configuration is highly unstable around the lateral axes, requiring active stability augmentation with the help of thrust-vectoring control and split flaps at the wing trailing edges. This is realised by automatically generated control laws based on Nonlinear Dynamic Inversion (NDI), see Fig. 7. This will not be further discussed in the frame of this contribution.

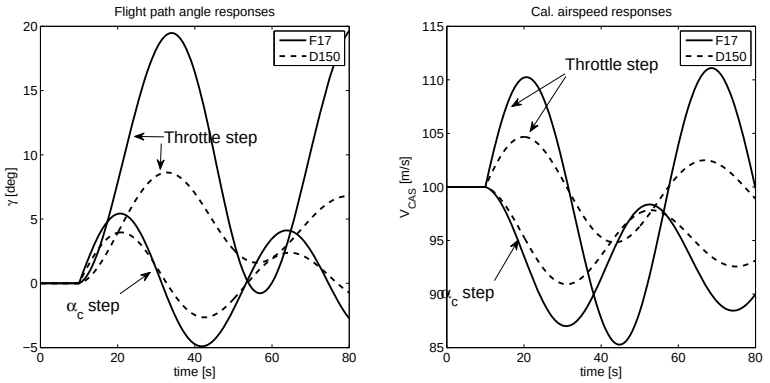

Fig. 8 Open loop longitudinal responses of the F17 (UCAV) and D150 (pass. aircraft) types to angle of attack and throttle step inputs

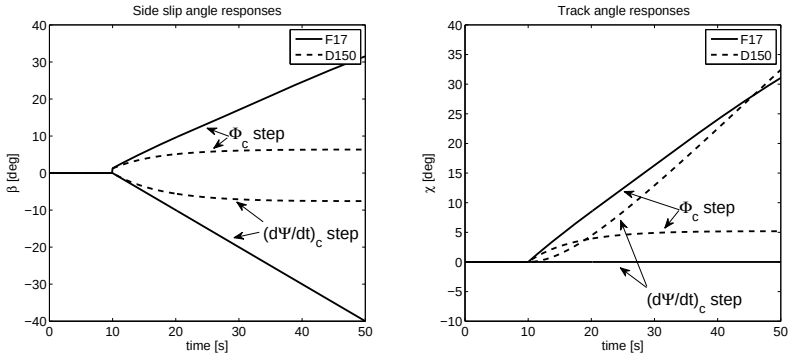

Fig. 9 Open loop lateral responses of the F17 and D150 configurations to roll angle and heading rate step inputs

4 TECS / THCS-based autopilot control laws

A mentioned before, the whole process from aircraft modelling, simulation set-up, simulation and computation of criteria of interest is fully automated and controlled via a database. This allows the use in for example Multi-disciplinary Design Optimisation (MDO) loops. Using autopilot control laws to fly the aircraft along the pre-scribed trajectory implies that these control laws must work reliably, regardless of type of aircraft. This has been one of the main reasons why the Total Energy (TECS) and Total Heading Control Systems (THCS) [10, 11, 13, 14] have been selected for this task. The *basic* control law structures are depicted in Fig. 10; for implementational details, the reader is referred to the references listed above and [6].

4.1 TECS

In TECS, thrust command computation is based on eqn. 5. Since aircraft mostly operate near their minimum drag speed, the main assumption is that drag changes slowly as a function of airspeed. As a consequence, thrust changes directly influence the specific excess power available for acceleration and flight path angle changes. For flight conditions where the assumption on drag is not valid, a proportional and integral control law ensures zero steady-state error in commanded speed and flight path angles (with $\sin\gamma \approx \gamma$):

$$\frac{\delta T_{com}}{mg} = \frac{K_{TI}}{s}\left(\frac{\dot{\hat{V}}_{err}}{g} + \gamma_{err}\right) - K_{TP}\left(\frac{\dot{\hat{V}}}{g} + \gamma_a\right) \tag{8}$$

Here $\dot{V}_{err}$ is the air-mass referenced acceleration error and γ_{err} is the flight path angle error. The speed error is computed in the active autopilot speed mode (mach, calibrated airspeed, true airspeed, ground speed). The flight path angle error in the active vertical path mode (altitude hold, vertical path tracking, etc.), see (see Fig. 5). This flight path angle is preferably also air-mass referenced, since otherwise the loop gain of altitude loops becomes strongly ground speed and therefore wind dependent.

TECS uses the elevator command channel to control the distribution of the specific excess power between rate of change in potential and kinetic energy. This is also done via a proportional and integral control law:

$$\delta\alpha_{com} = \frac{K_{EI}}{s}\left(-\frac{\dot{\hat{V}}_{err}}{g} + \gamma_{err}\right) - K_{EP}\left(-\frac{\dot{\hat{V}}}{g} + \gamma_a\right) \tag{9}$$

In order to obtain uniform command responses over the flight envelope, an angle of attack-command inner loop is used instead of commanding elevator directly. In this way, TECS commands the total lift, only requiring a simple correction proportional to the inverse of dynamic pressure to compensate for dynamic pressure effects on lift. In case of a 3 DOF model approximation, the angle of attack is a direct input variable. In case of a 6 DOF model, this command variable is to be controlled by an inner loop control law that simultaneously provides the required short-period damping.

As can be seen from eqn. (7), aircraft-specific parameters are mass, wing area and the lift curve slope C_{L_α}. These are easily compensated for in order to obtain uniform speed and flight path angle responses between different aircraft configurations.

4.2 THCS

The Total Heading Control System [14] controls the sum of heading angle Ψ and aerodynamic side slip angle β by means of the aircraft roll attitude angle:

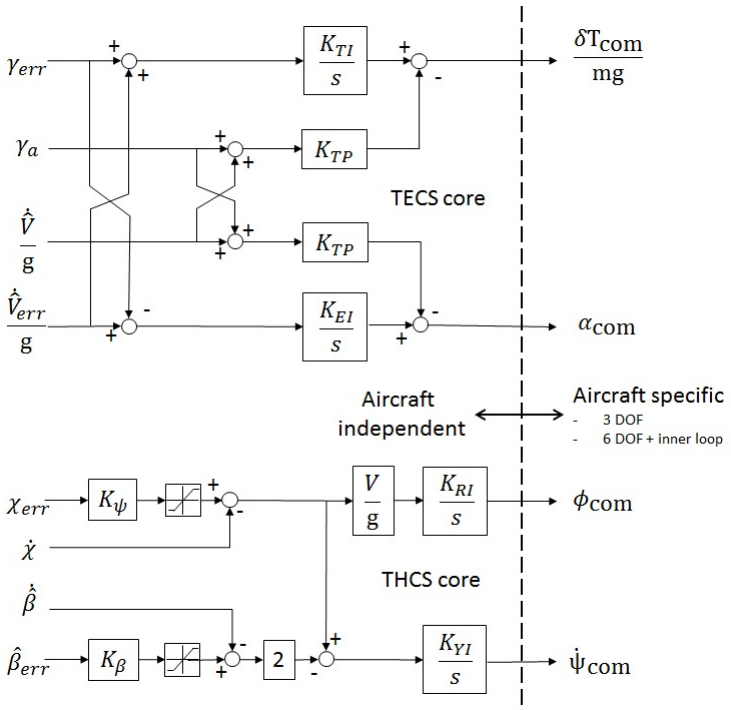

Fig. 10 Total Energy and modified Total Heading Control Systems

$$\Phi_{com} = \frac{K_{YI}}{s}\frac{V}{g}\left(K_\Psi \Psi_{err} - \dot{\Psi} + K_\beta \hat{\beta}_{err} - \dot{\hat{\beta}}\right) \tag{10}$$

where $\hat{\beta}$ is a side slip angle signal synthesised from air data and inertial measurements. The heading angle error Ψ_{err} is generated by autopilot modes for lateral navigation (course and lateral trajectory control). In case of cross wind components, the heading angle will be different from the inertial track angle χ due to the required crab angle. In order to avoid steady state path errors, the steady state crab angle may be subtracted from the heading angle command. During the projects REAL (EU-FP5 [17]) and Weather and Flying (DLR-internal [16]) DLR made good flight test experience with directly controlling the track instead of heading angle [16]. For this reason, THCS has been slightly adapted accordingly as compared with the original version depicted in [12], see Fig. 10. The track angle χ can be approximated by:

$$\chi \approx \Psi + \beta + \beta_w \approx \Psi + \beta_k \tag{11}$$

where β_w is the angle between the airspeed and inertial speed vectors, projected on the horizontal plane, and β_k is the flight path angle of side slip. The modified control law is defined by:

$$\Phi_{com} = \frac{K_{YI}}{s}\left(K_\Psi \chi_{err} - \dot{\chi}\right) \tag{12}$$

Note that, besides the β_w component, the control laws (10) and (12) are very similar. As with the original system, yaw rate $\dot{\Psi}$ is used for turn co-ordination:

$$\dot{\Psi}_{com} = \frac{K_{YI}}{s}\left(K_{\Psi}\chi_{err} - \dot{\chi} - 2(K_{\beta}\hat{\beta}_{err} - \dot{\hat{\beta}})\right) \tag{13}$$

The side slip angle feedback terms have been multiplied by two, since the use of track angle basically already includes side slip (eqn. (11)). The adapted structure less elegantly accommodates modes like de-crab in case of cross wind landing. However, in this application de-crab is performed by a separate control law. A very elegant feature of THCS is its handling of lateral cross wind shears. As will be shown in the following section, even a steadily increasing lateral wind gradient does not result in a steady state flight path error.

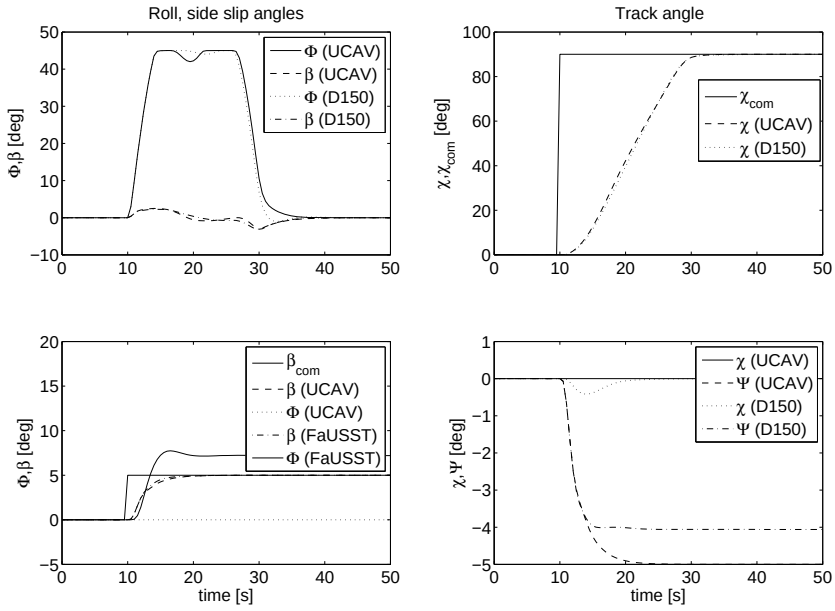

Fig. 11 Example simulations with dissimilar aircraft types (UCAV DLR-F17, D150). Upper plots depict track angle, lower plots side slip angle command responses

Figure 11 depicts step responses to track and side slip angle command inputs for both example configurations. As compared with Fig. 9, THCS clearly results in uniform command responses with completely identical gain settings. This is a key enabling factor for the fully automated mission simulation process. The same holds for the longitudinal TECS responses, see Fig. 12. Obviously, differences arise in the command inputs. For example, as the F17 UCAV model does not experience side

forces due to side slip, the commanded roll angle in case of a side slip command input remains zero (Fig. 11 lower left).

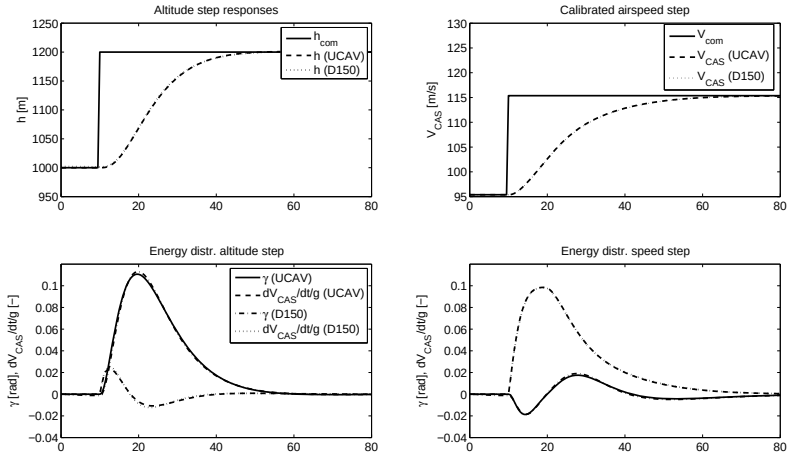

Fig. 12 Example simulations with dissimilar aircraft types (F17 UCAV, D150)

5 Application examples

The presented mission simulation environment and methodology have been used in various applications:

- Sensitivity study of basic geometry parameters of a short-range aircraft (D150 configuration) with respect to fuel consumption and gaseous emissions, based on a typical operational route between two airports;
- Sensitivity study of take-off climb and final approach descend profiles of a passenger aircraft with respect to noise impact on the ground (D150) [2];
- Evaluation of new approach procedures, like steep approaches and HeNAP (Helical Noise Abatement Procedure), see Fig. 4. Promising results of the latter resulted in an ATTAS flight test [3]. In this case, NDI-based inner loop control laws were implemented in combination with the same TECS control laws, see [16] and Fig. 7 (top). Interestingly, only the K_{EP} gain (eqn. (9)) had to be reduced since, instead of α_{com}, pitch attitude Θ_{com} was used as a command variable, see Fig. 10. The latter already provides phugoid damping due to implicit flight path angle feedback, since $\Theta \approx \alpha_k + \gamma_k$. This may also be a consideration in case of pitch channel saturation due to imposed angular limits in TECS: angle of attack control tends to immediately result in phugoid oscillations;

- Design optimisation of a stealthy unmanned combat aerial vehicle (UCAV). Mission simulation provided fuel consumption parameters during typical combat missions (see Fig. 3).

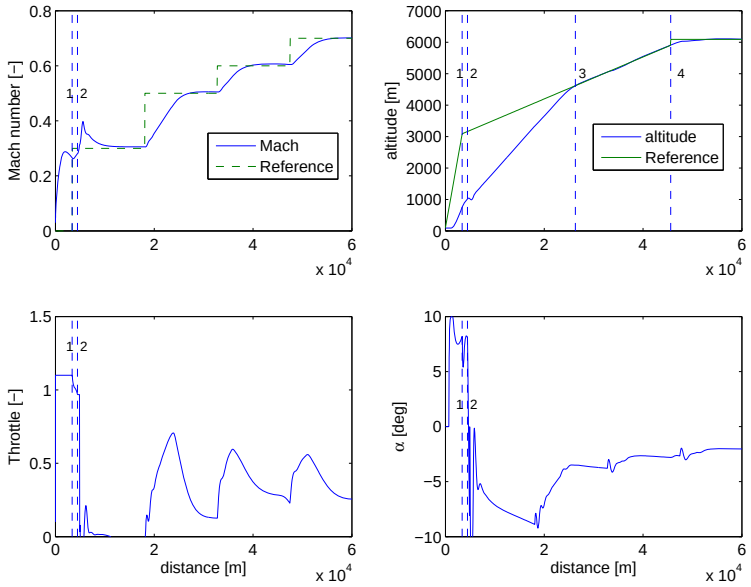

Fig. 13 Altitude profile of F17 UCAV from take-off to cruise

For the latter example, simulation results will be discussed in some more detail. The over-all mission is depicted in Fig. 3. The aircraft takes off and climbs out to a cruising altitude of 6000m. The Mach number and altitude profiles as well as throttle and angle of attack command inputs are depicted in Figure 13. The aircraft starts at zero speed and accelerates by applying full power (120 %). After rotating to a pitch angle of 10 degrees (lower right plot), the aircraft lifts off and the TECS/THCS-based autopilot is switched on (marked by the first vertical dashed line, 1). The autopilot immediately starts tracking the target mach number (dashed line). At the second dashed line, the aircraft flips on its back, showing its clean side (no landing gear doors, etc.) to the ground for lower radar visibility. This manoeuvre takes 4 seconds, during which the aircraft is attitude controlled only; the angle of attack is proportionally decreased to the negative of its value when starting the rolling manoeuvre. Note that the mach number increases and altitude decreases. After reaching inverted flight, the autopilot continues, tracking mach and altitude reference commands. It initially reduces throttle setting to zero to reduce speed to the commanded value (speed priority in TECS), then increases to track altitude at a given maximum climb rate (between lines 2 and 3). At line 4, the cruising altitude is smoothly cap-

tured. Note that after reversal, the angle of attack is necessarily negative all the time, increasing closer to zero as the mach number increases step wise (top left). This sign change obviously has to be made in the pitch channel of the TECS control laws as well.

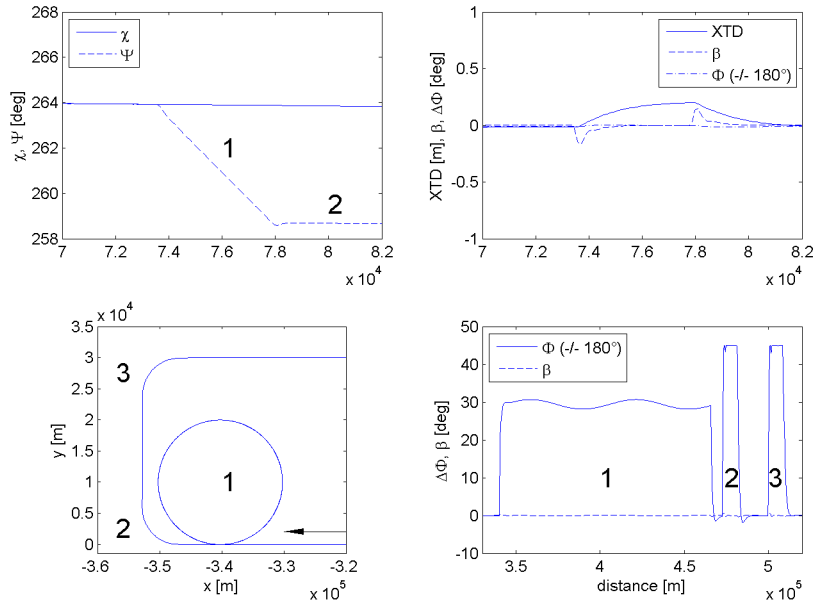

Fig. 14 Lateral cross wind shear during cruise and turning flight (XTD = cross track deviation, from Φ its value in straight and level flight of 180 deg has been subtracted for better visibility)

The two plots at the top of Fig. 14 depict a portion of the cruise segment. During this segment, a 20 m/s cross wind (from the left w.r.t. direction of flight) is introduced linearly in 20 seconds, introducing a lateral wind shear of 1 m/s^2. From the yaw attitude angle Ψ (top left) it can be seen that THCS rotates the aircraft into the wind in order to maintain its inertial course and keep cross track deviation (XTD) to zero. The lateral deviation is less than 30 cm, while side slip and roll angles remain very small.

The two plots below in Fig. 14 depict a circular holding pattern (two full turns, marked with 1) and two course changes to return to base (2 and 3). Due to the cross wind and maintaining a constant mach number, the roll angle is oscillating as function of the course angle due to varying ground speed. This is explained in detail in [16]. Note that, in spite of the cross wind, the top view of the trajectory (lower left) shows a single perfect circle. During the course changes (2 and 3), THCS commands maximum bank angle of 45 degrees (lower right plot). This again results in smooth circular course changes in the top view.

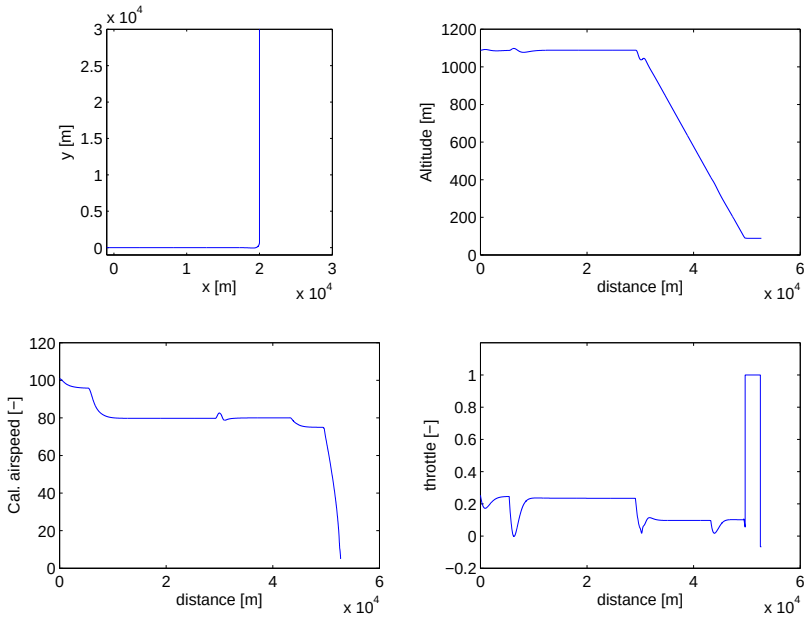

Fig. 15 Final approach, landing, and roll-out

Fig. 15 depicts the final approach and landing. The aircraft has already rolled back from inverted flight during descent from cruise flight. In the upper left plot (top view of trajectory), the runway is in the lower left hand corner. After the final right turn, the altitude decreases along a -3 deg glide slope. Landing touch down is followed by roll-out. Note that the altitude is barometric, so that its value ends at the field elevation (89 m in this case). The calibrated airspeed (lower left) is reduced to 75 m/s during final approach and than reduces towards zero after touch down. The aircraft performs a de-crab manoeuvre during flare, as the approach is flown with a cross wind (not shown). The throttle response (lower right) shows an increase to maximum thrust after touch down. This is combined with thrust reversal to help brake the aircraft.

This specific simulation has two goals: (1) compute fuel consumption for the given design configuration and (2) to provide trajectory data to specialist analysis tools for further analysis. One example is the computation of radar detection probabilities by the DLR Institute for Microwaves and Radar. The fuel consumption during this flight is depicted in Fig. 16. Total flight time is one hour and 24 minutes.

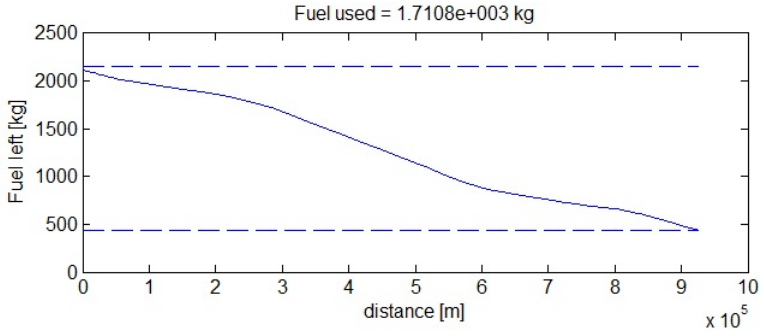

Fig. 16 Fuel as a function of flown distance

6 Conclusions

A new mission simulation environment has been presented. Since it is fully automated from model integration to computation of metrics of interest, it may be used in aircraft and engine optimisation loops. One challenge has been to make any given aircraft fly the specified mission in a consistent and repeatable way. This has been achieved by the use of autopilot control laws, based on the Total Energy and Total Heading Control Systems for vertical path, speed, and lateral path control. The control laws have turned out to work straight away with very diverse types of aircraft, without any modification of gains. In addition, in case the specified mission is partially not flyable (which may easily occur when its parameters are not selected carefully), protection features still result in consistent simulation results with a clear indication where deficits occurred.

References

1. Various Authors. PHARE : EFMS Phase 1B Technical Reference Document. Technical Report DOC 96-70-15, Eurocontrol, 1996.
2. Lothar Bertsch, Sebastien Guerin, Gertjan Looye, and Michael Pott-Polenske. The Parametric Aircraft Noise Analysis Module - status overview and recent applications. In *Proceedings of the 17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference)*, Portland, Oregon, USA, 2011.
3. Lothar Bertsch, Gertjan Looye, Eckhard Anton, and Stefan Schwanke. Flyover Noise Measurements of a Spiraling Noise Abatement Approach Procedure. *AIAA Journal of Aircraft*, 48(2), 2011.
4. Daniel Böhnke. CPACS – Common Parametric Aircraft Configuration Schema. Technical report, DLR presentation, 2011.
5. Rudolf Brockhaus. *Flugregelung*. Springer-Verlag, Berlin Heidelberg, 1994.
6. Nir Kastner and Gertjan Looye. Generic TECS based autopilot for an electric high altitude solar powered aircraft. In *Proceedings of the 2nd CEAS EuroGNC conference*, Delft, The Netherlands, April 2013.

7. Thiemo Kier, Gertjan Looye, Moriz Scharpenberg, and Marion Reijerkerk. Process, methods and tools for flexible aircraft flight dynamics model integration. In *Proceedings of the International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, 2007.
8. Thiemo Kier, Gertjan Looye, Moriz Scharpenberg, and Marion Reijerkerk. Unifying Manoeuvre And Gust Loads Analysis Models. In *Proceedings of the International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, 2009.
9. Andreas Klöckner, Martin Leitner, Daniel Schlabe, and Gertjan Looye. Integrated Modelling of an Unmanned High-Altitude Solar-Powered Aircraft for Control Law Design Analysis. In *Proceedings of the 2nd CEAS EuroGNC conference*, Delft, The Netherlands, April 2013.
10. A A. Lambregts. Operational aspects of the integrated vertical flight path and speed control system. *AIAA Paper 83-1420*, 1983.
11. A A. Lambregts. Vertical flight path and speed control autopilot design using total energy principles. *AIAA Paper 83-2239*, 1983.
12. A.A. Lambregts. Aircraft Automatic Control Systems Design, 1996. Lecture notes on 5 seminars given at Delft University of Technology, The Netherlands.
13. Antonius A. Lambregts. TECS Generalized Airplane Control System Design An Update. In *Proceedings of the 2nd CEAS EuroGNC conference*, Delft, The Netherlands, April 2013.
14. Antonius A. Lambregts. THCS Generalized Airplane Control System Design. In *Proceedings of the 2nd CEAS EuroGNC conference*, Delft, The Netherlands, April 2013.
15. Gertjan Looye. The new DLR Flight Dynamics Library. In *Proceedings of the 6th Modelica Conference*, Bielefeld, Germany, 2008.
16. Gertjan Looye. *Helical Flight Path Trajectories for Autopilot Evaluation*, pages 79–90. Springer, 2011.
17. Gertjan Looye and Hans-Dieter Joos. Design of autoland controller functions with multi-objective optimization. *AIAA Journal of Guidance, Control, and Dynamics*, 29(2):475–484, March–April 2006.
18. Gertjan H.N. Looye. *An Integrated Approach to Aircraft Modelling and Flight Control Law Design*. 2008. Doctoral thesis Delft University of Technology. Available from `http://repository.tudelft.nl`.
19. Modelica Design Group. Modelica: Language design for multi-domain modeling. `http://www.modelica.org`.
20. Reiko Müller. Multi-Objective Optimization of an Aircraft Trajectory between Cities using an Inverse Model Approach. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, Minneapolis, Minnesota, August 2012.
21. Reiko Müller. A constrained inverse modeling approach for trajectory optimization. In *Submitted to the AIAA Modeling and Simulation Technologies Conference*, Boston, MA, August 2013.