Aircraft Mission Simulation with the updated FlightDynamics Library

Marc May[1] Reiko Müller[1] Gertjan Looye[1]

[1]DLR Institute of System Dynamics and Control, Oberpfaffenhofen, Germany,
marc.may@dlr.de, reiko.mueller@dlr.de, gertjan.looye@dlr.de

Abstract

Aircraft mission simulation plays an essential role during the simulative design process of aircraft systems. An important task is to conduct analyses for total aircraft assemblies in different flight phases and also to produce performance metrics in Multidisciplinary Design Optimization (MDO) loops. Based on DLR's FlightDynamics, the library presented in this paper introduces mission simulation capability for aerial and on-ground movement. For instance, this includes flight control and management functions, a detailed aircraft implementation including several subsystems (for example landing gears, actuation and sensor systems), and the application of the integrated setup to realistic use-cases.

Keywords: *Aircraft modeling, mission simulation, flight control.*

1 Introduction

During design and testing of aircraft and associated systems, the need for aircraft mission simulation arises in many occasions, for example to conduct MDO, validation and verification, trajectory optimization and prediction, as well as air traffic management. The main building blocks of a mission simulation setup are on the one hand the actual aircraft model with its various (sub-) systems, and on the other hand functions and algorithms (mostly in the form of controllers) enabling the aircraft to follow a mission plan (for example a gate to gate trajectory for a passenger aircraft). Concerning the first part, a well established tool to perform aircraft modeling in Modelica is DLR's FlightDynamics library (Looye et al. 2014), which has been used in a multitude of research projects so far. The second part, namely flight control and management algorithms allowing the guidance on ground and in the air was added to the FlightDynamics during the course of the EU-funded project OPeRATOR, which is associated with Systems Integrated Technology Demonstrator (SYS-ITD) of the CleanSky2 Joint Undertaking (MISSION Consortium 2015). Aside from DLR, partners of the OPeRATOR project (including a large aircraft supplier) seeked a library with this capability to help developing and assessing aircraft subsystems, for example novel actuator architectures.

The mission simulation capability necessitated several adaptations and further development of the FlightDynamics library, which are laid out in this paper. The following section reviews the derived library structure, highlighting changes with respect to the original FlightDynamics. In Section 3, the modelling of the aircraft and associated systems will be explained, followed by the mission and scenario definitions in Section 4, where also the flight controller and flight management systems necessary to guide the aircraft on the mission are described. Results of the use-cases defined in the project for verification are shown in Section 5, and the conclusions are drawn in Section 6.

2 Library structure

As a derivative of the FlightDynamics library, OPeRATOR shares its high level library structure, which is shown in Figure 1. It furthermore adds a dedicated Controller package, which contains all mission planning and execution capabilites (i.e. controllers and flight management algorithms).

The first level subpackages will be shortly listed in the following. In UsersGuide, a text-based tutorial of the library, its components and modeling principle is given. Also, there is a list of references given in the Literature model. The Examples package implements the use-cases defined in the project, results of these are given in Section 5. MyProject is used to include and integrate new aircraft designs. Therefore basic building blocks needed to assemble a model, like aerodynamics, engines and systems are stated here, which represents a proposal how to structure a new aircraft implementation with new systems. FlightVehicle is the main location where all models and classes are stored. This comprises equations of motion in the kinematics package, implementations of aerodynamics (a tabulated dataset from Vortex Lattice Method (VLM) calculations is used in OPeRATOR, however arbitrary analytical and data-driven models are possible), engine and actuator models, sensors, landing gears, and so on. New models are added to the respective sub-library when they are designed. The Environment package contains everything necessary for simulating the surrounding of the aircraft, for example geoid, gravity and

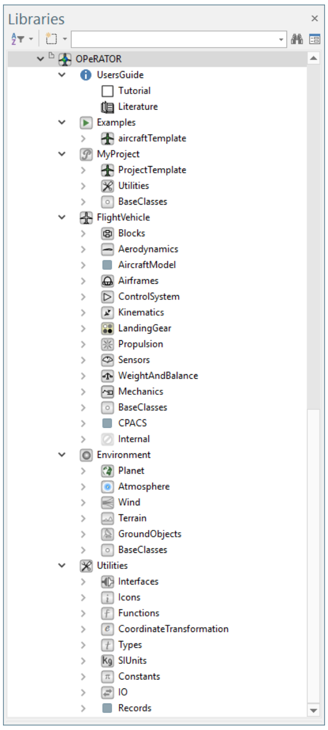
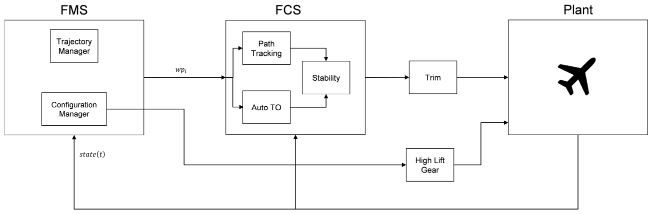
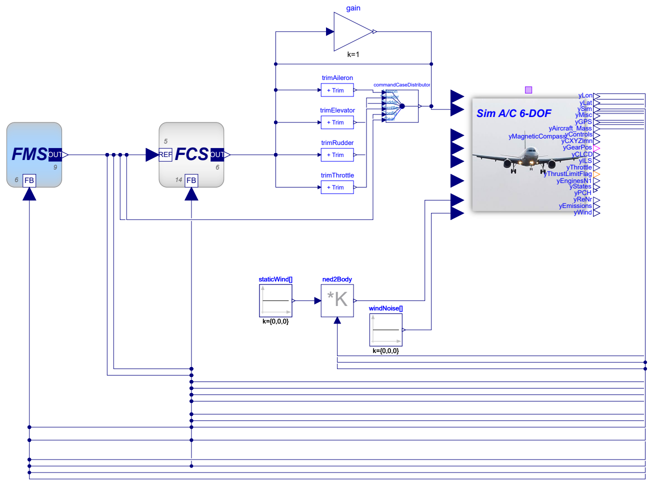


Figure 1. The library tree opened in SIMULATIONX® (this version contains a hidden controller package, which is mandated by export control regulations for dissemination to external partners).



(**a**) Mission simulation architecture.



(**b**) MODELICA implementation.

Figure 2. Layout of the flight management and control systems in conjunction with the aircraft model for mission simulation.

magnetic flux of the Earth, atmosphere and ground objects like localizer and glideslope emitters. The final package Utilities contains interface definitions, coordinate transformation and various helper functions and records. Finally, the Controller package collects all functionality concerning flight control. This encompasses autopilot functionality (e.g. acquiring and holding of reference altitude, speed and orientation), trajectory path tracking, controllers for stabilization and damping of aircraft eigenmodes and disturbances as well as controllers for specialized tasks like takeoff and landing.

3 Aircraft modeling

Depending on the use-case or application, an aircraft modeling library must provide different levels of fidelity or detail. In the case of aircraft mission simulation, reduced mass-point models were the standard approach so far, as they combine fast execution time and acceptable precision. Well-known implementations include for example the Base of Aircraft Data (BADA) model family (EUROCONTROL 2012) as a reduced longitudinal-plane aircraft performance model, and also inverse models with three Degrees of Freedom (DoFs), like they were used for instance in the DLR project TIVA (Liersch and Hepperle 2011). For the OPERATOR project, a six DoF rigid body aircraft model formulation was selected as a generic approach, which enables a more detailed simulation of transient flight, especially during turning.

The equations of motion (see e.g. Stevens, Lewis, and Johnson (2015) for a derivation) with introduction of forces and moments, as well as coordinate transformations are implemented using the MODELICA MULTIBODY library (Otter, Elmqvist, and Mattsson 2003), which allows straightforward graphical construction of airframe assemblies with components, like propulsion, actuation and sensor systems as well as payloads. Furthermore, the library employs partial models for all subcomponents based upon which different realisations of systems can be developed and exchanged very easily. Global models related to world/geodetics and atmosphere are adopted from the DLR ENVIRONMENT library (Briese, Klöckner, and Reiner 2017), and use the inner/outer concept of MODELICA. The overall architecture of the resulting mission simulation setup is shown in Figure 2, with some of the system components being described in the following sections, namely aerodynamics (3.1), actuators (3.2) and landing gears (3.3).

3.1 Aerodynamics

The aerodynamics model in the OPERATOR library works on tables previously generated by a multidisciplinary design and simulation network, for which the

DOI 10.3384/ecp21181155

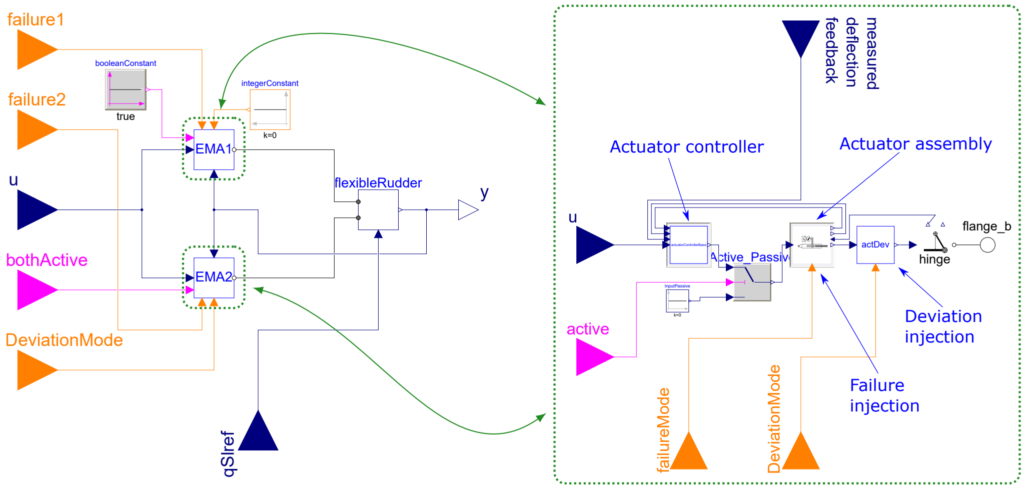


Figure 3. Aileron actuation system with redundant Electro-Mechanical Actuator (EMA) implementation.

MDO software Remote Computing Environment (RCE) (Seider 2014) is employed. The tools for each discipline of the multidisciplinary design are incorporated in RCE and are made available by several DLR institutes, each being specialized in a different area of aircraft design. The interconnected execution inside RCE allows design workflows adapted to the respective use-case (for example aircraft noise analysis and CFD/CSM simulation). The RCE workflow has already been used in different projects, for instance DIGITAL-X (Kroll et al. 2016) and VICTORIA (Görtz et al. 2020).

One of the results of the data calculated within RCE are aerodynamic tables, derived for example from Computational Fluid Dynamics (CFD) data. Regarding mission simulation and the large time horizons involved, the aerodynamics model has to balance the aspects of fidelity and precision versus computation time. To this end, for modeling of drag and lift contributions of the airframe and control surfaces, a fast VLM method (Hedman 1966) is combined with a lifting-line method (Horstmann 1986), yielding tables scheduled over different sets of variables from altitude, Mach and Reynolds numbers, angle of attack and sideslip angle, as well as body turn rates and control surface deflections.

As the scenario considers passenger aircraft trajectories, extreme maneuvering or conditions at the boundary of the flight envelope are not expected. This means that these linear methods are applicable and provide a good approximation of the aerodynamics during the entire mission. For modeling the increase in lift coefficient c_L due the ground effect, several empirical relations stated in Phillips and Hunsaker (2013) are implemented, which can be selected as a parameter in the aerodynamics model.

3.2 Actuation system

The actuation system can model physical actuators (for example by employing MULTIBODY library parts) as well as approximations like first and second order filters. Furthermore, a combination of approximated and physical actuators can be used. To demonstrate these functionalities, a use-case was defined regarding an actuator force-fight as introduced in more detail in Section 5.1. In this scenario, one aileron actuator is assumed to be defective leading to a force-fight with a second actuator which operates on the same flexible control surface. For this purpose, a redundant actuator setup with two Electro-Mechanical Actuators (EMAs) per control surface was developed as shown in Figure 3. The deviations are adjusted via an integer variable that triggers the following cases:

- Gain difference in the actuators, which leads to divergence in positions and rates.
- Time delay, meaning that the actuators react at different time instants.
- Position offset, i.e. the actuators extend to different positions when subjected to the same command input.

In a similar way, it is possible to introduce failures of the actuators, such as a stuck actuator or actuator runaway with a parameterized rate up to a maximum / minimum deflection.

3.3 Landing gear

The aircraft's undercarriage is represented by a simplified landing gear model assembly consisting of two main and one nose gear (see Figure 4). All gears are suspended by a spring-damper element. The two main gears feature brakes, while the nose gear is not braked and has a steering. By means of a MULTIBODY frame connector (Otter, Elmqvist, and Mattsson 2003), the gear is attached to the airframe. In order to facilitate numerical calculation of tyre friction and general forces and moments of

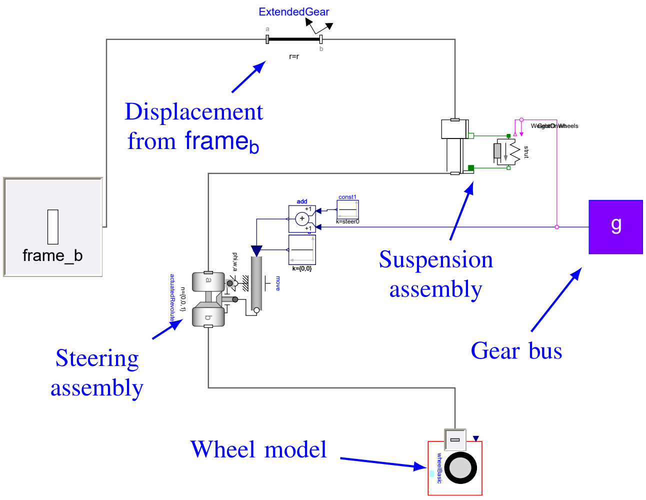


Figure 4. Steerable nose gear implementation. The connector frame_b is attached to the aircraft reference point in the top-level airframe assembly.

the gear assembly, the wheels are defined to remain on the ground-projected local coordinate system below the aircraft during the complete flight. Forces and moments are only introduced to the system if the landing gear assembly touches the ground as represented by the altitude-triggered boolean variable `WeightOnWheels`. A tyre friction model has been adopted from Otter, Elmqvist, and Mattsson (1999), which considers a hybrid formulation for the different states (forward- and backward rolling, stuck, sliding) for Coulomb friction in x- and y-direction of the tyre local coordinate system. If the friction elements are dynamically coupled, event definition for these states leads to a mixed continuous/discrete system of equations, that has to be solved by the numerical integration algorithm. Finally, an empirical roll-coefficient model according to Barnes and Yager (1998) has been implemented which allows to consider different runway surface conditions (dry, wet, flooded, icy, snow-covered) depending on loading, tyre pressure, braking threshold and external conditions. This overall approach allows to efficiently handle the unavoidable event iterations for the discrete landing gear touchdown, and provides good estimates for the forces and moments involved.

3.4 Model validation and verification

As OPERATOR is based on the FLIGHTDYNAMICS library, several components have been validated and verified during its commercialization process (for example the environment models, kinematics, aerodynamics, engine, weight and balance, actuator, sensor, terrain and wind modules as well as interfaces). Nevertheless, model verification was performed at hand of five use cases (controlled flight in cruise, actuator force fight, automatic landing, rejected takeoff, and full city-pair mission) in OPERATOR. Data like aerodynamics and propulsion maps were generated by verified processes and tools within the DLR by the institutes specialized in the respective topics (for example the Institute of Propulsion Technology[1]). In CLEANSKY2, the OPERATOR library is also employed by industrial partners (e.g. Collins Aerospace) in combination with their own proprietary models, where additional validation and verification activities are performed accordingly.

4 Flight management and control

It is assumed in the OPERATOR project, that the general mission definition is stated beforehand and subsequently provided to the mission simulation tool. The respective input file specifies a gate-to-gate mission separated into several phases, as it is most common practice in Air Traffic Management (ATM) and flight planning.

Table 1. Mission simulation input data.

Mission point ID	Segment duration [s]	Altitude [ft]	Mach number
Ramp Up	900	0	0
Taxi	900	0	0.025
Takeoff	60	0	0.2
Climb	180	4000	0.4
Level flight	1020	39000	0.6
Level flight	2700	39000	0.78
Descent	1320	2350	0.6
Approach and flare	120	0	0.25
Landing	60	0	0.025
Taxi	480	0	0.025

4.1 Mission definition

Within the phases, values for several variables are defined which parameterize the segment (see Table 1) and which may also serve as parameters for an outer optimization loop. According to these definitions, a variable denoting the current flight state is synthesized, which in turn controls the modes and settings in the Flight Management System (FMS). In MODELICA, this is realized by the enumeration `FlightState`, which can assume the values `UN` – Undefined (this is the initial state, from which transition to other states can happen, see Figure 5), `RU` – Aircraft Ramp Up, `TA` – Taxi phase, `TO` – Takeoff phase, `CL` – Climb phase, `CR` – Cruise phase, `AP` – Approach phase and `LD` – Landing phase.

The switching between flight states is realized with a state machine (see Figure 5) and depends on several boolean conditions (grey source blocks) which trigger the transitions between states. The conditions are determined by variables such as the weight on wheels, the distance

[1] `www.dlr.de/at/en/`

DOI 10.3384/ecp21181155

to target runway, altitude-, orientation-, rates, airspeed and corresponding limits. The simulation is usually terminated if the landing is completed and the aircraft has stopped, but other exit conditions can be defined easily.

4.2 Flight management

To discretize the trajectory with the `FlightState` variable, it needs to be processed initially. In Figure 6, the overall process of generating an FMS solution from the textual mission input data using the Trajectory manager is shown. By taking into account the target runway position (latitude φ_{rwy}, longitude λ_{rwy}, altitude h_{rwy}) and heading χ_{rwy} as well as tabulated time intervals Δt, altitudes h and velocity/Mach number Ma, the trajectory is back-propagated in position starting at the touchdown point. During this initialization, the initial trajectory point is moved into the touchdown point, and direction is rotated by 180°. A loop over all segments is performed, which allows to calculate the positions of transition points (also in time) and the ground distance (on great circle paths) between points. This finally yields the departure conditions in position and initial heading. The trajectory is hence represented by linear segments where additional transient phases are considered to allow smooth transition between segments and flight states. The user can influence these transitions by specifying maximum accelerations in longitudinal and vertical directions. By taking into account the state feedback x_{feedback}, continuous commands in the variables altitude h_{cmd}, pitch angle γ_{cmd}, inertial velocity V_{cmd}, course angle χ_{cmd} and sideslip angle β_{cmd} are provided for the current flight state.

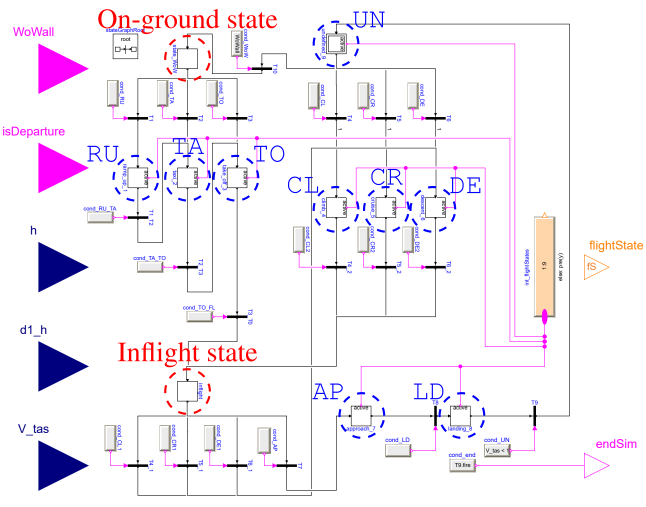


Figure 5. The state machine for switching of flight states using the MODELICA STATEGRAPH library. From the initial `UN` state, it can transition either to on-ground (`RU`, `TA`, `TO`) or inflight-states (`CL`, `CR`, `DE`).

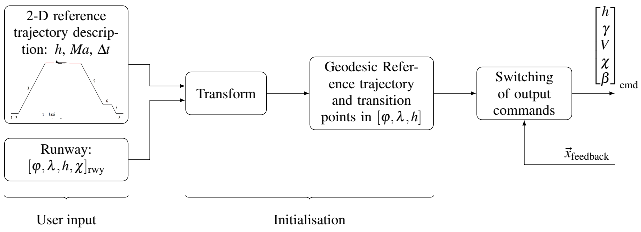


Figure 6. Transformation of mission input data using the Trajectory manager.

4.3 Flight controller

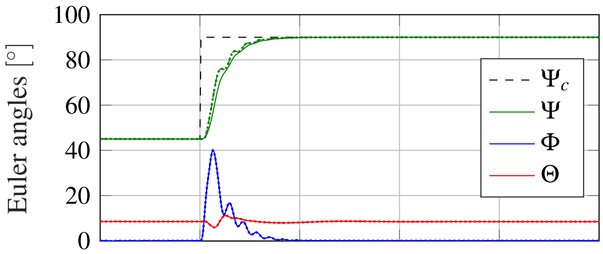
For the Flight Control System (FCS), the common cascaded controller structure with autopilot outer- and stabilization inner loop is adopted (see Figure 2a). The path tracking module receives the aforementioned continuous commands from the FMS and contains modes for inflight and on-ground movement, while a dedicated block is developed for automatic takeoff. These two modules form the autopilot part of the Flight Control System (FCS), which issues aircraft orientation and throttle commands to the stabilization loop. Automatic landing is performed with the guidance by Instrument Landing System (ILS) and Localizer sensor models, which provide lateral and vertical deviations to the glideslope controller. It also contains a mode for the landing flare before touchdown, which works according to the variable τ - law described in Lambregts (1982).

5 Simulation examples

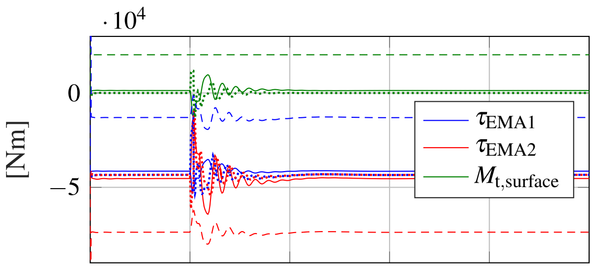
In this section, the results of the use-cases defined in the OPERATOR project are presented. The use-cases include the force-fighting scenario in Section 5.1, the rejected takeoff scenario in Section 5.2, and the complete mission in Section 5.3. All scenarios have been tested with SIMULATION-X® version 4.1 and DYMOLA® 2018 FD01 on a standard PC-workstation (INTEL XEON E5-1630 v3, 16 GB RAM) with WINDOWS 10. The real time factor in DYMOLA® is varying (force-fight: 328.9, rejected takeoff: 26.4, mission: 213.5) for this configuration and the considered scenarios, which is due to different initialization states (initialization on ground with extended landing gear, or with extended actuator model is more complicated and takes more time), the number of states and the system stiffness.

5.1 Actuator force fight

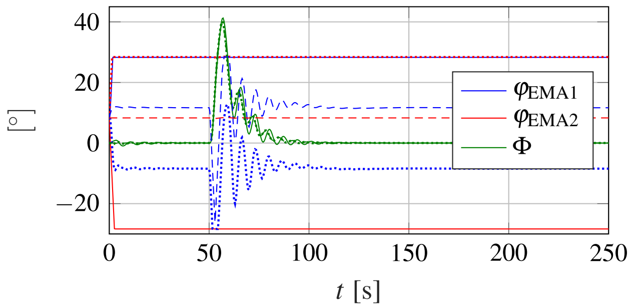
The inflight force-fighting scenario has the goal of evaluating the aircraft's behaviour with respect to control inputs and possible faults. To this end, the implementation considers an actuator force fight scenario, where redundant aileron actuators allow to model different types of force fights. These are possible differences in actuator gain, and time delay, as well as deflection offsets which results in deformation of the flexible control surface.



(a) A course change of 45° is initiated at $t = 50$ s upon which the malfunctioning EMA2 counteracts the healthy EMA1. Line styles correspond to the different deviation cases, see Figure 7b below.



(b) The case of actuator gain deviation is denoted by solid lines (——, ——, ——), while for time delay by (········, ········, ········) and position offset by (- - - -, - - - -, - - - -).



(c) .The actuator failure cases are denoted by (——, ——, ——) for an upwards runaway, (········, ········, ········) for a downwards runaway and a stuck actuator is indicated by (- - - -, - - - -, - - - -).

Figure 7. Results of the actuator force fight simulation.

Results for deviations of the actuator gain of +25%, a time delay of 0.2 s and a 10 cm actuator rod position offset are shown in Figure 7b, where always the second actuator (EMA2) introduces the deviation. During a course change (in this case 45°), the aircraft has to build up a roll angle for turning flight and reduce it again to return to wings level (the roll angle limit is set to ±45° inside the lateral autopilot). The actuators twist the flexible aileron control surface via torques τ_{EMA1}, τ_{EMA2} and hence introduce a torsional moment $M_{t,surface}$ (indicated by green color) that is different for the three deviation types. The difference in the overall gain of the actuator controller for instance produces a nonzero twisting moment (——), while the moment due to a time delay (········) vanishes if there is no control activity (i.e. when the new course is attained). The position offset in EMA2 - - - - is introduced after initialization/trimming in order to make the three solutions comparable. EMA1 - - - - has to produce a countering torque (symmetrical to the trim value, which is approximately represented by ········ and ········), which leads to a large twisting torque - - - - in the control surface.

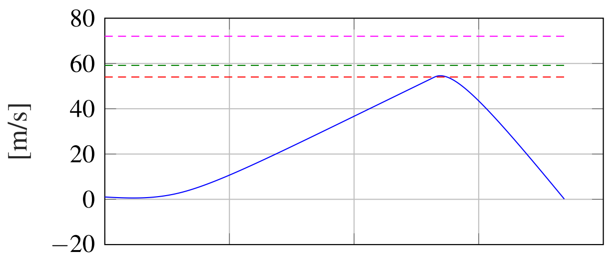
In Figure 7c, simulation results for three actuator failure types are depicted. It has to be noted that the aileron control surface can deflect upwards with larger angles than downwards before reaching limits/stops. Hence runaways of EMA2 with a rate of 0.25 m/s show, that for the downwards direction EMA1 ········ compensates the runaway of EMA2 ········ and still has control authority left during the turn at $t = 50$ s. This cannot be accomplished in the upwards runaway case, as due to the smaller downward deflection, EMA1 —— cannot cancel EMA2 —— anymore and therefore does not contribute to turning and also needs compensating moments from other aileron control surfaces. The final stuck case is similar to the downwards case, as EMA2 - - - - remains at a relatively low deflection, which can be cancelled by EMA1 - - - -. As can be seen in Figures 7a and 7c, the resulting Euler angles are very similar, with only small differences in the roll angle Φ and heading Ψ.

5.2 Rejected takeoff

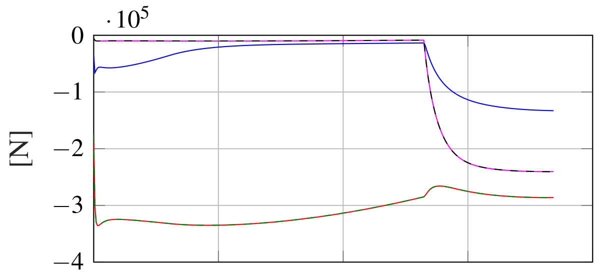
The rejected takeoff is a certification relevant test where the aircraft is required to stop in a certain amount of distance while the takeoff procedure is already underway. Due to the high speeds of around 300 km/h, a large amount of kinetic energy has to be converted to heat by the braking system, additionally in a short amount of time and taking into account the limited runway length. Depending on these factors and also parameters like ground roll coefficients, payload, on-ground controller etc., the decision speed beyond which no safe stopping is guaranteed, varies accordingly.

Representing an emergency maneuver, a dedicated braking mode is activated in the controller, while standard FMS and longitudinal FCS loops are switched off. This mode issues brake and reverse throttle commands (if supported by the engine model) to minimize the braking distance. In the simulation, this is triggered if the on-ground velocity surpasses the decision speed (this marks the point where in the case of problems the aircraft shall still be able to stop safely without taking off). To ensure smooth transitions, the velocity (0 m/s) and braking commands (100%) are filtered before being passed to the FCS. The lateral on-ground mode also acts to keep the aircraft at the centerline via the lateral autopilot and inner loops commanding the rudder and nose-wheel steering with a speed-variable gain (see Figure 8c). In the second subplot, the resulting longitudinal and vertical forces on the tyres of the nose and main gears are shown. As the aircraft accelerates, the nose wheel is unloaded (——) and the main gears are loaded in z-direction (—— and - - -) due to the pitch-up moment, in-

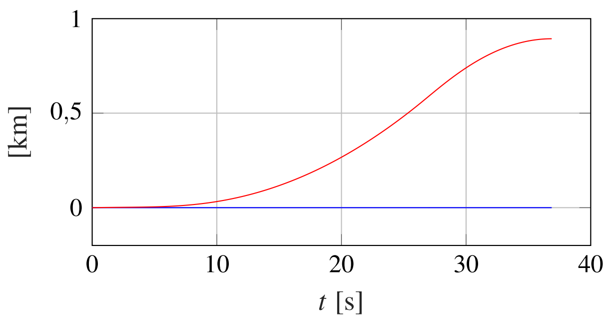
DOI
10.3384/ecp21181155



(a) The velocity —— is increased until decision speed - - - -, upon which the braking is initiated before rotation speed - - - - (climbout speed is given by - - - -).



(b) x- and z-forces for main gear right (—— and ——) , main gear left (- - - and - - -), as well as nose gear z-force ——.



(c) Distance travelled on ground —— and lateral deviation from runway centerline ——.

Figure 8. Results of the rejected takeoff simulation.

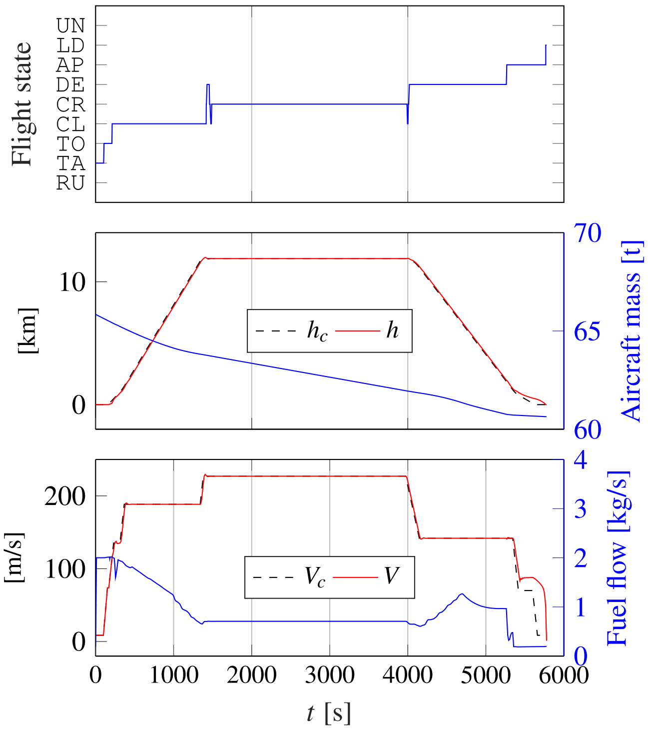


Figure 9. Results of the mission simulation.

duced by the engine to Center of Gravity (CoG) moment arm. Conversely, the main gear x-forces (—— and - - -) increase when brakes are applied, and the nose gear z-force increases due to the pitch-down moment during braking. Note that the z-values do not return to their stationary values, as the simulation is terminated as soon as the velocity falls below 0.1 metres per second. With a total mass of 72.69 t, the aircraft is able to stop after a distance of 882.5 metres (——), including the acceleration phase.

5.3 City pair mission

As described in Section 4, a standard city-pair mission is generated from the data supplied by the user. This includes climb, cruise and descent phases, which are flown in the longitudinal plane, i.e. except for departure and approach, there is no lateral manoeuvring. In the simulation and optimization studies targeted by the MISSION project, the aircraft has to climb for instance from ground altitude to cruise altitude and can do so in a variety of trajectories that are optimal for the functioning of specific systems and the mission length. When introducing additional system models, such as an Environmental Control System (ECS), the effects of environmental conditions (e.g. temperature, humidity) on dedicated aircraft systems can be studied. Similarly, during descent, the aircraft has to descend from cruise altitude to approach altitude, and the (auto-)pilot has to reduce the engine throttle. Again studies regarding ECS and engine operational modes can be conducted.

These prerequisites motivate the presented longitudinal approach in the main flight phases and a full 3-D simulation during departure and approach, for example to correctly simulate an ILS guidance. In Figure 9, the commanded and actual altitude h_c and h as well as velocities V_c and V are shown. Due to the flight controller, the aircraft can follow the reference trajectory with only small deviations. One exception is during approach, where the mission definition commands a trajectory that is higher and faster than what the aircraft is capable to achieve. The controller adjusts the commands so that the throttle is reduced to idle (see fuel flow) and the descent rate is at the current maximum value. The final approach is performed using the ILS system and flare controller as described in Section 4.3.

6 Conclusion and outlook

In this paper, recent extensions to the FLIGHTDYNAMICS library enabling passenger aircraft mission simulation and specialized case studies have been presented. Through integrated flight management and control systems, it is possible to conduct a variety of simulation studies, for example to design and evaluate subsystems like actuators, controllers, Environmental Control Systems (ECSs), landing gear systems and so on. This is showcased by means of three examples, an inflight actuator force-fight, a rejected takeoff simulation and a complete city-pair mission scenario.

In the future, the library will be continuously extended and employed for further studies and developments at the DLR, for example in the fields of unmanned and autonomous flight, simulation of future propulsion concepts (electrical, hydrogen, synfuels) as well as design and validation of control systems.

Acknowledgements

The work in project OPERATOR was funded by the CLEANSKY2 joint undertaking under topic number JTI-CS2-2017-CfP07-SYS-03-16.

References

Barnes, Arthur G. and Thomas J. Yager (1998). “Enhancement of aircraft ground handling simulation capability”. In: *Advisory Group for Aerospace Research and Development (AGARD)*.

Briese, Lâle Evrim, Andreas Klöckner, and Matthias Reiner (2017). “The DLR Environment Library for Multi-Disciplinary Aerospace Applications”. In: *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*. 1 132. Linköping University Electronic Press, pp. 929–938.

EUROCONTROL (2012). *User manual for the Base of Aircraft Data (BADA) Revision 3.10*. Technical/Scientific Report No. 12/04/10-45. EEC.

Görtz, Stefan et al. (2020-09). “Ergebnisse des DLR-Projekts VicToria - Virtual Aircraft Technology Integration Platform”. In: *Deutscher Luft- und Raumfahrtkongress 2020 (DLRK2020)*. URL: https://elib.dlr.de/135959/.

Hedman, Sven G. (1966). *Vortex lattice method for calculation of quasi steady state loadings on thin elastic wings in subsonic flow*. Tech. rep. DTIC Document.

Horstmann, Karl-Heinz (1986). “Ein Mehrfach-Traglinienverfahren und seine Verwendung für Entwurf und Nachrechnung nichtplanarer Flügelanordnungen”. PhD thesis. TU Braunschweig.

Kroll, N. et al. (2016-03). “DLR project Digital-X: towards virtual aircraft design and flight testing based on high-fidelity methods”. In: *CEAS Aeronautical Journal* 7.1, pp. 3–27. ISSN: 1869-5590. DOI: 10.1007/s13272-015-0179-7.

Lambregts, Antonius A. (1982). “Avoiding the pitfalls in automatic landing control system design”. In: *AIAA Guidance and Control Conference*. American Institute of Aeronautics and Astronautics. Chap. 6, pp. 799–809. DOI: 10.2514/6.1982-1599.

Liersch, Carsten M. and Martin Hepperle (2011-12). “A distributed toolbox for multidisciplinary preliminary aircraft design”. In: *CEAS Aeronautical Journal*. CEAS Aeronautical Journal Volume 2.Number 1-4, pp. 57–68. URL: https://elib.dlr.de/74509/.

Looye, Gertjan et al. (2014-09). “Object-Oriented Aircraft Modelling with the DLR FlightDynamics Library”. In: *63. Deutscher Luft- und Raumfahrtkongress*. Augsburg, Germany: Deutsche Gesellschaft für Luft- und Raumfahrt.

MISSION Consortium (2015). *MISSION Requirements Document and Exploitation Plan, Clean Sky 2 SYS-ITD Deliverable, D100.3.1.2*.

Otter, Martin, Hilding Elmqvist, and Sven Erik Mattsson (1999-09). “Hybrid Modeling in Modelica based on the Synchronous Data Flow Principle”. In: *IEEE International Symposium on Computer Aided Control System Design*, pp. 151–157. DOI: 10.1109/CACSD.1999.808640.

Otter, Martin, Hilding Elmqvist, and Sven Erik Mattsson (2003). “The new Modelica Multibody Library”. In: *3rd International Modelica Conference*, pp. 311–330.

Phillips, W. F. and D. F. Hunsaker (2013). “Lifting-Line Predictions for Induced Drag and Lift in Ground Effect”. In: *Journal of Aircraft* 50.4, pp. 1226–1233. DOI: 10.2514/1.C032152.

Seider, Doreen (2014-11). “Open Source Framework RCE: Integration, Automation, Collaboration”. In: *4th Symposium on Collaboration in Aircraft Design*. URL: https://elib.dlr.de/93323/.

Stevens, Brian L., Frank L. Lewis, and Eric N. Johnson (2015). *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons.