The New DLR Flight Dynamics Library

Gertjan Looye
German Aerospace Center (DLR), Institute of Robotics and Mechatronics
Oberpfaffenhofen, 82234 Wessling, Germany

Abstract

An overview of the new Modelica Flight Dynamics Library of the German Aerospace Center DLR is given. This library is intended for construction of multi-disciplinary flight dynamics models of rigid and flexible flight vehicles. The environment models provide the functionality to cover on-ground operations up to flight at high speeds and high altitudes. The resulting models may be used in various fields and stages of the aircraft development process, like flight control law design, as well as for real-time flight simulation.

Keywords: flight dynamics; flight control; simulation; aeroelasticity

1 Introduction

Dynamic simulation plays an important role in the aircraft design and certification process. Typical examples are development of flight control laws, flight loads analysis, specification and testing of on-board systems, aircraft handling qualities and system assessment in real-time manned flight simulators. DLR has developed an extensive Modelica library that allows for construction of suitable aircraft flight dynamics models for the various stages and applications in the aircraft development process. The following strengths of Modelica have hereby been exploited particularly:

- *Multi-disciplinary modelling:* multi-disciplinary interactions play an important role in flight dynamics. Especially when flight control laws are involved, aspects like flight mechanics, structural dynamics, and systems may show considerable dynamic interactions that must be appropriately addressed in the model used for design analyses. Modelica provides an ideal basis to develop such models, since a large amount of discipline-specific libraries is available that may be used to construct components of the integrated model;
- *Single source modelling:* Especially in flight controls various types of analysis methods are used that require aircraft models to be available in various forms. Examples are nonlinear models for simulation, (symbolic) linear models for stability and robustness analysis, inverse models for control law synthesis, etc. Various forms of models often involve independent implementations of the same model data. Modelica allows for the construction of a single model from which appropriate analysis models may be generated with the help of a model translator.

Back in 1995 the DLR institute of Robotics and Mechatronics developed a first library (at the time, based on the Dymola language) for modelling of aircraft flight dynamics [26]. Objective was to build a solid basis for constructing integrated dynamic aircraft models, including flight dynamics, detailed on board system dynamics, structural dynamics, etc. First applications were a generic transport and a fighter aircraft [20] and a first flexible aircraft shortly thereafter [15]. Since then, the library has been expanded and applied to complex aircraft models that include e.g. system hydraulics and electronics [25, 27, 24]. In the frame of the international projects REAL (Robust and Efficient Autopilot control Laws design, funded by the EU in the fifth frame work programme [31]) and the GARTEUR action group AG-11 on clearance of flight control laws [11], the automatic generation of inverse models for fast trim computation and nonlinear control laws was applied for the first time. Recent application examples are the thrust-vectored X-31A high-angle of attack experimental aircraft and a real-time capable integrated flight dynamics and aeroelastic transport aircraft model, including unsteady aerodynamics, structural dynamics, control system, etc.

The latest version of the Flight Dynamics Library includes various major enhancements as compared with previous versions. Firstly, it exploits more recent Modelica features, like the bus concept and the matured inner-outer concept. Secondly, the library is now compatible with the Multibody Library (part of the Modelica standard library), allowing for easy construction of airframes. Thirdly, its *world* model has been enhanced to comply with inertial standards like WGS84 [4] for position reference.

This paper discusses these latest developments, new features, and example applications of the Flight Dynamics Library. In the following section the selection of a generic aircraft model structure is discussed (Section 2), based on which the Flight Dynamics Library has been organised (Section 3). In Section 4 the automatic generation of model code for model simulation and analysis is discussed, followed by some example applications. Finally, a summary is given in Section 6.

2 The aircraft model structure

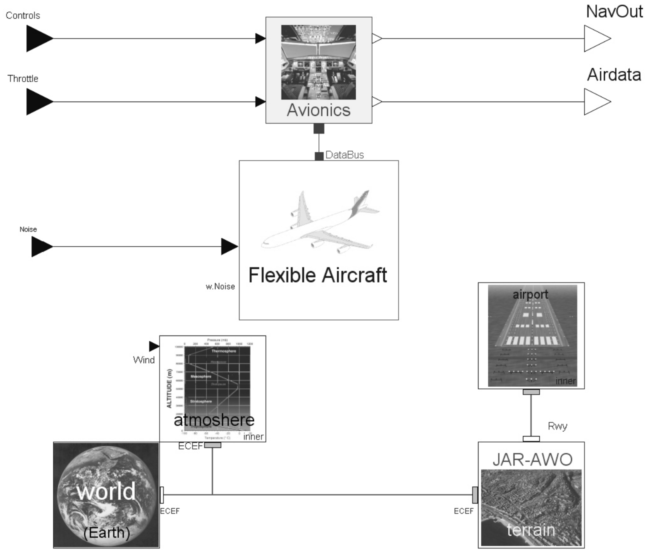

Figure 1: Top-level of model: aircraft and environment

The objective of this section is to introduce and motivate the basic structure of an aircraft model as it may be composed from the Flight Dynamics Library. The structure of this library will be discussed thereafter.

In constructing complex models the choice of hierarchy is crucial, since this largely determines how model components interact. For the Flight Dynamics Library a top-level model structure as shown in Figure 1 has been adopted. It consist of one or more aircraft, and environment objects (The *avionics* component and input and triangular-shaped output connectors will be discussed in Section 4). The environment objects include a *world*, *atmosphere*, *terrain*, and *airport* model. Note that the (in this case, single) aircraft model has no direct link with the environment models, which physically makes sense. Using the so-called inner-outer feature of the Modelica language, these models provide field functions. For example, the aircraft may request its surrounding atmospheric conditions from the atmosphere model by sending its local inertial position. Any other aircraft (or e.g. sensor) object in the model may do this as well. This is an advantage as compared with most block-oriented libraries, where an atmospheric model is directly linked to, and thus occupied by, the one aircraft. The ability to easily include multiple air vehicles is useful for applications involving mutual interactions, like towed gliders, wake vortexes, air-to-air refuelling, release of missiles, etc.

2.1 The world model

In the following subsections the environment models in Figure 1 (*world*, *atmosphere*, *terrain*, *airport*) will be discussed. These components determine validity of the over-all model to a large extent. Most important is the *world* model (in this case the Earth, but the underlying base class may be extended to implement models of other planets), since it provides the inertial reference in the form of the so-called Earth-Centred Inertial (ECI) reference frame. Its origin is attached to the Earth's centre of mass, its orientation is fixed with respect to reference stars. In addition, the model component has the following functions:

- Provide the geodetic reference. As indicated in Figure 1, to this end the World Geodetic System 1984 [4] (WGS84) is used. The object implements an Earth-Centred Earth-Fixed (ECEF) reference frame, which has the same origin as the ECI, but rotates with the Earth. The attitude of the ECEF (w.r.t. ECI) is available in a connector. A set of functions transform ECI and ECEF referenced position vectors into geodetic longitude, latitude, and height co-ordinates (w.r.t. WGS84 ellipsoid) and vice versa. For a given longitude and latitude, another function provides the local undulation of the so-called EGM96 (Earth Gravitational Model 1996) geoid with respect to the WGS84 ellipsoid, providing the Mean Sea Level (MSL) reference [14].

- Implement a model of the Earth's gravitation. The gravitational model to be used with WGS84 is the Earth Gravitational Model 1996 (EGM96), provided in the form of tables describing equipotential surfaces a function of longitude and latitude. Currently, a more simplified height and geocentric latitude-dependent (Ref. [33] -

Eqn.(1.4-16)) and a constant gravity model are available.

- Implement a model of the Earth's magnetic field. This field is required to compute indications of compass models. The model is based on the US National Geo-spatial-Intelligence Agency (NGA) World Magnetic Model (WMM), which is published every five years and predicts the time-varying intensity and direction of the magnetic field as a function of WGS84 longitude, latitude and height. The current model covers 2005 till 2010 [23].

Double-clicking on the *world* object in Figure 1 allows a number of parameters to be set, like whether the Earth is rotating or in rest, initial day time, and the type of gravity model (approximate EGM96, height independent, or constant). The features of the object may be overkill for many applications, but provide sufficient generality for use with for example high speed and high altitude flight vehicles. Furthermore, the applied WGS84 ensures compatibility with standard GPS equipment, with most flight simulator vision systems, navigation system models, etc. Obviously, any parameter set in the *world* and other environment models applies to all components in the aircraft model.

2.2 The atmosphere model

The second environmental object in Figure 1 is the atmosphere. Normally, the International Standard Atmosphere (ISA) as a function of the height above MSL is used. Alternatively, parameters for constant atmospheric conditions may be entered. The air mass is nominally assumed to be in rest with respect to the ECEF, explaining why a connection with the *world* ECEF-connector exists. However, the component also foresees implementation of wind fields. Currently, wind components in northern and eastern directions may be entered at a reference altitude of 100 ft above the Earth surface. A simple Earth boundary layer model logarithmically reduces the wind velocity to zero on the ground.

2.3 The terrain model

To the right in Figure 1 a terrain model has been added. A component containing highly detailed, or simple parametrised models of the Earth's surface may be selected from the library. Depicted in Figure 1 is a terrain model as used for automatic landing control law design and certification, based on EASA CS-AWO specifications [10]. The location, elevation, direction, and slope of a runway may be specified, as well as slopes and steps in the terrain below the approach path. A simple function call from e.g. an aircraft sensor then returns the corresponding local terrain elevation above MSL or the WGS84 ellipsoid, allowing for computation of for example the reading of the radio altitude sensor.

2.4 The airport infrastructure model

The airport object implements earth-fixed navigational equipment (e.g. VOR, DME, ILS systems at specified locations). In the figure the ILS equipment of the one runway as positioned in the EASA CS-AWO terrain model is included. Specific characteristics like glide slope angle and antenna transmitter positions may be specified via parameters. Any other model object may obtain its local glide slope and localiser deviation via a simple function call.

2.5 Rigid and flexible aircraft models

The core of the model structure is of course the component that represents the actual aircraft. The Flight Dynamics Library foresees the implementation of rigid just as well as flexible ones. A typical model structure for a flexible transport aircraft is shown in Figure 2. The components resemble physical parts an aircraft consists of (airframe, engines, actuators, sensors), and phenomena it is influenced or driven by (kinematics, aerodynamics, wind).

Component interconnections

For mechanical interconnections the connectors from the Multibody standard library [28] are used. For use with flexible airframes an extended version has been implemented that includes generalised co-ordinates and forces. For the engines and sensors the connector represents the point at which the device is attached to the airframe. Aerodynamic forces actually act all over the airframe. However, in flight mechanics usually only the summed effect with respect to some reference point, like the Aerodynamic Centre (AC), is of interest. Also for the aerodynamic forces in the aeroelastic aerodynamic model (to be discussed shortly) in fact only their summed effect is considered due to so-called left-generalisation with rigid and flexible eigenmodes, see Ref. [19]. Therefore, the aerodynamic models need a single connector only.

Kinematics

The backbone of the model depicted in Figure 2 are the

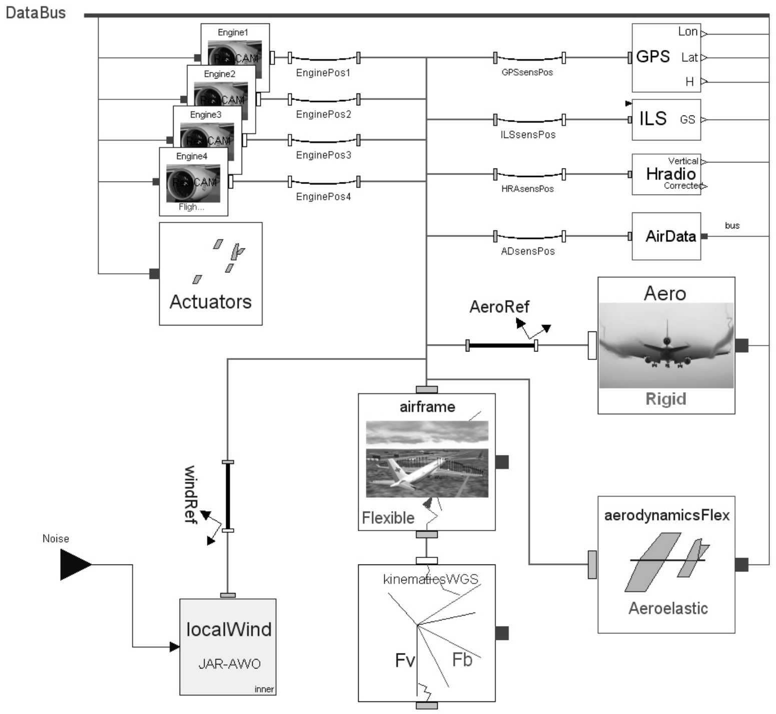

Figure 2: Structure of the *Flexible Aircraft* model in Figure 1

kinematics and *airframe* components. The first defines a “North-East-Down” (NED) local vertical frame with its origin moving with a fixed position in the aircraft, preferably the centre of gravity. The object also defines a right-handed body-fixed reference frame with its origin at the same location, but with a fixed attitude w.r.t. the airframe (x-axis towards the nose, z-axis down). The attitudes and inertial positions of both reference systems are available in the two connectors. The one on top represents the aircraft body reference system, the one belows represents the vehicle-carried NED reference frame.

Airframe
The difference between a rigid and a flexible aircraft is, in fact, only in the *airframe* object. In case of a rigid airframe, it contains the standard Newton-Euler force and moment equations with respect to a body reference system [7] (attitude and position in lower connector). Although the origin of this reference system is preferably the centre of gravity (for compatibility with standard flight dynamics models), a fixed point w.r.t. the undeformed airframe shape may be more useful for referencing reasons. The local gravity acceleration is obtained by a call to the *world* object (Figure 1). Note that the computation of gravity depends on the method that is selected in the *world* object. In case of a flexible airframe, linear elastic equations of motion in modal form augment the Newton-Euler equations [34, 30]. The body axes system is hereby considered as a so-called mean axis system. The momentary shape of the airframe is characterised by states in the form of generalised co-ordinates (also called mode shape multipliers). The underlying data (modal mass, damping, stiffness, and mode shape matrices) are automatically read from a specified file prior to simulation. More details on nonlinear equations of motion of flexible aircraft can be found in Refs. [19, 30].

Connection of the *airframe* object to the *kinematics* object (see Figure 2) makes that the reference systems in both connectors merge, i.e. from then on the airframe is moving freely with respect to the inertial reference, according to kinematic equations described in the *kinematics* object.

The *airframe* object has a second connector on top (see Figure 2). This connector may contain a different reference frame with a constant offset, or may simply be identical to the body frame (to be specified via an offset parameter). It is intended for interconnection of for example external force model components, sensor models, etc. As mentioned before, in case of a flexible airframe also generalised co-ordinates and generalised forces are included in the connector definition.

External forces and moments
The airframe equations of motion are primarily driven

by aerodynamic and propulsion forces and moments. These are computed in corresponding model components in Figure 2. These components often need to be prepared for each aircraft type individually, since application rules and data (sources) behind aerodynamics and propulsion models may strongly differ. For this reason, a base class is available that already defines interfaces and the connector, as well as equations for computation of key variables like the angle of attack, side slip, and true airspeed. Local wind velocities are hereby requested from the *atmosphere* model in Figure 1. The user may develop own model components, inheriting this base class.

Besides the airframe, each component may be developed around its own local reference frame. In case of aerodynamics, these may for example be the stability or wind axes. Interconnection with the airframe follows via a transformation object (e.g. *AeroRef* in Figure 2). This object has two connectors representing two reference systems. The offset (position, orientation) in between may be specified via parameters that become visible and can be edited by double-clicking on the object. The object also relates the forces and moments that act along the connector reference systems. When connecting a model with the *airframe* object, the transformation object makes sure that the kinematics between the local component and the airframe reference systems are correctly related, as well as forces and moments are applied correctly.

The aircraft model in Figure 2 has two aerodynamics models (right hand side). The upper one (*Aero*) contains forces and moments as induced by the over-all motion of the aircraft (“rigid aerodynamics”), usually also corrected for quasi-steady deformation of the airframe. The underlying model may be based on complex application rules, table look-ups, etc. In case a data set is not available or incomplete, computational tools as described in [13] are used. The lower aerodynamics component (*aerodynamicsFlex*) computes unsteady (generalised) forces and moments as induced by flexible deformation of the airframe. For this component extensive pre-processing tools have been developed, involving application of the Doublet-Lattice Method, axis transformations, Rational Function Approximation and removal of quasi-steady effects (already accounted for in the rigid aerodynamics model), see Refs. [19, 13] for more details. The unsteady aerodynamic data are read from a user-specified data file at simulation start.

Note that the *Aero* component is connected to the lower *airframe* connector via the *AeroRef* object, whereby the latter describes the offset between the airframe body axes and the aerodynamic reference system. The upper aerodynamics component is directly connected with the upper *airframe* connector, making use of generalised co-ordinates declared therein. In case kinematics and the balance between aerodynamic and actuation forces are relevant, a direct interconnection between the *actuators* and aerodynamics models may be added.

The engine models (top left) are connected to the airframe via a slightly different type of transformation. Instead of an offset, the number of a structural grid point, where the object is to be attached, may be specified. At simulation start the transformation object requests the rows of the modal matrix that apply to the grid point from the *airframe* object, allowing it to continuously compute the kinematic relation and force balance between its connectors as a function of the offset from the airframe reference and the local deformation [19, 30]. This for example implies that directional thrust variations due to local deformation at the engine attachment point are automatically taken into account.

Sensor models

The very same principle as used for interconnecting engine models with the airframe structure also applies to the sensor models, located in the top-right corner of Figure 2. A set of sensor types is available in the library. For example, accelerometers compute local accelerations at their point of attachment (specified via grid point number, or offset) as a function of the inertial motion of the airframe, its position in the airframe reference, as well as the local airframe deformation.

The ILS, GPS, and radio altimeter sensors obtain their values by making a function call to the *airport*, *world*, and *terrain* environment models respectively (Figure 1), passing on their momentary inertial position as an argument. In this way, for example multiple GPS sensor objects may be included at various locations on the airframe. Each object can request its very local co-ordinates from the *world* object.

Local wind effects

As already discussed in Section 2.2, mean winds are computed in the atmosphere block at the top level of the model in Figure 1. However, turbulence models are usually described in aircraft body axes, whereby delays as gusts travel along the airframe, are taken into account. This is described in the *localWind* object (lower left, in this case based on EASA CS-AWO specifications for autoland assessment). Random turbulence velocities are obtained from dedicated filters

(Dryden, Karman) that use white noise signals as inputs. This noise is provided via an external connector.

Systems
On-board systems are included in the *actuators* component. This component may describe actuators and hydraulic / electric systems using simple transfer functions, as well as highly detailed physical models, constructed from hydraulics and electronics libraries. The library currently only provides the first variant, since detailed on-board system models are unique for each type or family of aircraft and are usually provided by systems specialists. A recent example of on-board system model implementation using Modelica can be found in [5].

Avionics bus
Finally, the thin bar at the top of Figure 2 represents a so-called *data bus*, implemented using Modelica's expandable connector concept. The data bus includes signals that one would typically find on avionics buses in the aircraft, like the readings of all sensors, command signals to engine and control surface actuators, gear status, etc. For this reason, the sensor, actuator, and engine models have been attached to the bus object. The bus is also accessible from outside and allows direct connection to elements from the Modelica block diagram library. This enables a control system composed using this library to directly communicate with the aircraft data bus.

3 The Flight Dynamics Library

The top-level structure of the DLR Flight Dynamics Library is depicted in Figure 3. The *Modelica* branch in the depicted tree (top) contains the principal standard libraries delivered with Modelica. The branches of the Flight Dynamics Library will be briefly described below:

- **ProjectTemplate** contains a basic library structure for an aircraft. Each aircraft type has its own models for aerodynamics, propulsion, systems, landing gears, etc. These models are built on base classes that already compute all basic variables (e.g. for aerodynamics, angle of attack, calibrated airspeed, etc.) and are stored in this structure. The project template contains a very simple, but readily working aircraft model. The user may copy this template into an own project and start implementing aircraft-specific components, or add components from the Flight Dynamics or any other Modelica library.
- **Aerodynamics** contains example aerodynamic models for use in rigid and flexible aircraft models, as well as base classes that the user may extend (inherit) to develop his own aircraft-specific model components. Each aircraft type or family namely tends to use unique application rules. For this reason, newly implemented aerodynamics components are stored within the aircraft project (see ProjectTemplate).
- **Airframes** contains rigid and flexible airframe model objects. The rigid ones may have constant mass and inertia tensor (entered via parameters), or these may change at a given rate (e.g. as a function of fuel consumption). The flexible airframe component loads its mass, and modal data from an external file (e.g. a Matlab mat-file [21]).
- **Environment** contains all environment-related models as described at the beginning of Section 2.
- **Examples** contains example implementations of various (basic) aircraft models.
- **Gear** currently contains a simplified landing gear model for which basic properties may be set and which may be attached to the *airframe* object in Figure 2. A base class containing a standard interface for interconnection with the *airframe* is provided for implementation of detailed landing gear models, e.g. composed with help of the multibody library by specialists in the field.
- **Interfaces** contains all library-specific connector types, as well as the data bus that was discussed Section 2.5.
- **Kinematics** contains the *Kinematics* object as described in the previous section.
- **Propulsion** contains, as for the aerodynamics, example engine model implementations, as well as base classes that allow the user to implement his own propulsion models.
- **Systems** mainly contains sensor models (accelerometers, ILS, GPS, etc.) with time constants and noise if desired, and simple transfer function-based actuator models.
- **Transformations** contains standard transformations between reference systems.

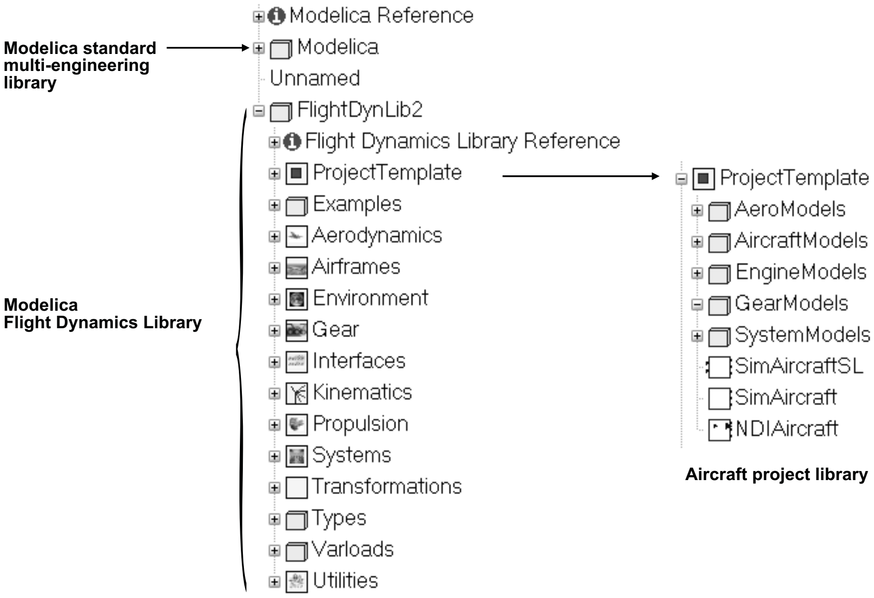

Figure 3: Top-level structure of the Flight Dynamics Library

- **Types** contains type definitions for internal variables, to which the user may add his own.

- **Utilities** contains miscellaneous functions e.g. for reading external data files.

Within a dedicated Modelica modelling and simulation environment, like Dymola (Dynamic Modelling Library [3]) aircraft models may be composed from the library using drag and drop.

4 Automatic code generation

After model composition has been finished, a model translator sorts and solves all model equations according to specified inputs and outputs into Ordinary Differential Equations (ODE's) or Differential Algebraic Equations (DAE's), suitable for use in simulation. A modelling tool that is well capable of doing this is Dymola [3]. Besides a graphical modelling environment and advanced symbolic algorithms, the tool offers extensive simulation and data analysis capabilities. However, the model code may be used in other engineering environments and simulation tools as well, like for example Matlab/Simulink [22]. For this environment an additional tool set has been developed that automatically generates trimming and linearisation scripts, allowing the user to easily specify and accurately compute initial conditions prior to simulation [15].

4.1 Specification of inputs and outputs

A simple way of specifying model inputs and outputs is shown at the top of Figure 1. Here a so-called Avionics block has been connected to the bus connector of the aircraft. At this main model level, also input and output connectors have been defined. The Avionics block injects pilot throttle and control surface input commands (from Throttle, Controls connectors) into the data bus. Output variables of interest, in this case navigation and air data, are read from the bus and passed on to output connectors (*NavOut*, *Airdata*).

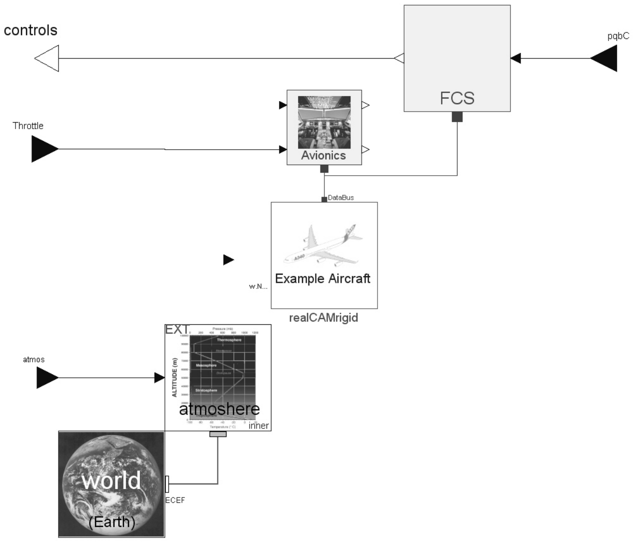

Figure 4: Reversal of inputs and outputs for NDI control law generation

4.2 Inverse model generation

Probably one of the most attractive features of Modelica, in combination with a model compiler like Dymola, is the possibility to generate inverse models just as easily as normal simulation models. Inverse models are extremely useful for fast and accurate trim computation, systems and control surface sizing, as well as for automatic generation of control laws that are based on inverse model equations, like Nonlinear Dynamic Inversion (NDI [9]). Ref. [18] describes various types of inverse model-based control laws and their automatic generation from Modelica models. Figure 4 shows the addition of an FCS block, which contains the basic structure for an NDI controller, see Figure 5. Like the *Avionics* block, the FCS communicates with the aircraft model via the expandable bus. Compared with Figure 1, the *Controls* input has become an output, and command variables *pqbC* (roll rate p_b, pitch rate q_b, side slip angle β) have been added as inputs instead. This is basically all that is needed to generate an inverse model. For practical reasons, additional modifications have been made, like the removal of most environment models: variables like radio altitude and ambient pressure can be directly obtained from measurement instead.

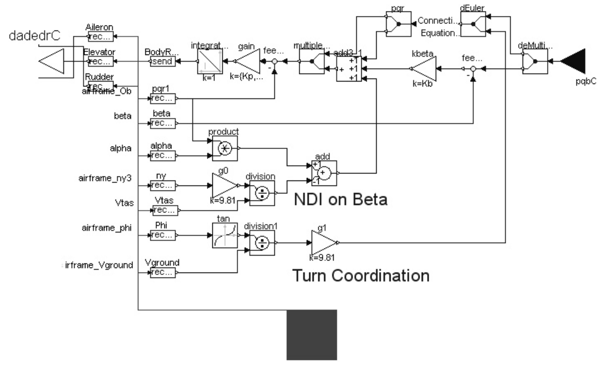

Figure 5: NDI control law with command variables for manual aircraft control

4.3 Desktop visualisation

Easy access to high-quality desktop visualisation tools becomes more and more important in the flight control law design process. This helps the engineer to better understand the (closed-loop) dynamics of the aircraft, and allows her or him to interactively "fly" the aircraft to qualitatively assess control law performance and to find weaknesses before implementation in the full flight simulator. One of the first commercial environments offering this visualisation capability is the Aviator Visual Design Simulator (AVDS), described in Ref. [29]. An interface with the Flight Dynamics Library has been described in [24]. Interfaces with public domain simulator programs like FlightGear [2] are to be developed, whereby the internal flight dynamics model in this program is overwritten by the model constructed from the Flight Dynamics Library.

5 Application examples

Since its first version in 1995, the Flight Dynamics Library has been applied in several projects at DLR, especially involving model development for design and evaluation of flight control laws. A number of these applications will be briefly discussed in this section.

REAL – Automatic Landing
In the frame of the EU-project REAL (Robust and Efficient Auto pilot control Laws design [31]) for the first time inverse model equations for a transport aircraft were automatically generated from the model implementation in Modelica. These inverse equations were used as the core of an automatic landing system that was developed in the frame of this project [16]. The control laws were successfully flight tested on DLR's test bed ATTAS (Advanced Technologies Testing Aircraft System [6]) during six automatic landings.

X-31A with reduced vertical tail
The same procedure for automatic generation of Nonlinear Dynamic Inversion control laws was applied to the thrust-vectored experimental fighter aircraft X-31A in the frame of the project VECTOR (Vectoring, Extremely short take-off and landing, Control, Tailless Operations Research [12]), in order to investigate reduced vertical tail configurations of this aircraft [32]. The control laws were exported from Dymola and implemented in the ground-based flight simulator in Patuxent River, MD, USA, and successfully evaluated by five test pilots and one fleet pilot.

3D-Flexible aircraft flight simulator
From the example flexible aircraft model as presented in Section 2 simulation code was generated for use in interactive real-time simulation. The model was augmented with automatically generated Dynamic Inversion-based control laws, the automatic landing system as developed in the project REAL, as well as load alleviation control laws. An engineering visualisation environment called *VisEngine*, developed by AeroLabs AG [1], simultaneously visualises aircraft flight dynamics and structural dynamics in real-time

in very high quality, see Figure 6. As exclusive features for DLR, VisEngine was extended with on-line visualisation of airframe deformation, as well as the capability of visualising the aircraft and its environment using 3-D stereo projection. The visualisation of structural deformation greatly helped to qualitatively assess performance of structural control laws.

GARTEUR FM(AG17) Control law design for aircraft on ground
In the future, transport aircraft may be equipped with drive-by-wire control laws to reduce pilot workload during landing roll-out, taxiing, and take-off. In addition, this technology will provide the basis for development of functions for automatic ground manoeuvring [8]. In the frame of this GARTEUR project a complex aircraft model, including landing gears, was implemented using the Flight Dynamics Library. In order to study various possible command variables for the pilot, automatic model inversion was used to develop control laws based on nonlinear dynamic inversion. More details can be found in Refs. [19, 17].

Figure 6: 3D-stereo visualisation of aircraft flight and structural dynamics, and aerodynamic loads

6 Conclusions

In this paper an overview of the Modelica-based Flight Dynamics Library has been given. This library allows for intuitive construction of multi-disciplinary models for use in the aircraft and flight control laws design process. To this end, the library offers the following unique features:

- full compatibility with other libraries based on the Modelica language, allowing for development of truly multi-disciplinary aircraft models on a common modelling platform;
- intuitively structured models due to a physics-oriented break-down into model components and interactions;
- construction of rigid just as well as fully flexible aircraft models, including unsteady aerodynamic effects;
- easy implementation of multiple aircraft models using the same set of environment models (earth, atmosphere, etc.);
- automatic generation of efficient simulation code for various engineering environments (using a Modelica tool like Dymola);
- automatic generation of inverse model code, e.g. for use in nonlinear control laws;
- automatic generation of trimming and linearisation scripts for use with the model in Matlab/Simulink;
- easy integration with desktop visualisation.

References

[1] Aerolabs AG – Professional Solutions for the Engineering Industry: `http://www.aerolabs.de`.

[2] Flight Gear web site: `http://www.flightgear.org`.

[3] Dynasim AB. *Dymola – User's Manual*. `http://www.dynasim.se/`, 1994–2006.

[4] anon. Department of Defence World Geodetic System 1984 – Its Definition and Relationships with Local Geodetic Systems, Third Edition, Amandment 1. Technical Report NIMA TR8350.2, National Imagery and Mapping Agency, Bethesda, MD, January 2000. Third Edition.

[5] J. Bals, G. Hofer, A. Pfeiffer, and C. Schallert. Virtual Iron Bird – A Multidisciplinary Modelling And Simulation Platform For New Aircraft System Architectures. In *DGLR Luft- und Raumfahrtkongress 2005, Friedrichshafen, DGLR-Jahrbuch 2005*, 2005.

[6] M. Bauschat, W. Mönnich, D. Willemsen, and G. Looye. Flight Testing Robust Autoland Control Laws. In *Proceedings of the AIAA Guidance, Navigation and Control Conference 2001, Montreal CA*, 2001.

[7] Rudolf Brockhaus. *Flugregelung*. Springer-Verlag, Berlin Heidelberg, 1994.

[8] J. Duprez, F. Mora-Camino, and F. Villaumé. Control of the Aircraft-on-Ground Lateral Motion During Low Speed Roll and Manoeuvers. In *IEEE Aerospace Conference Proceedings*, pages 2656 –2666, 2004.

[9] Dale Enns, Dan Bugajski, Russ Hendrick, and Gunter Stein. Dynamic Inversion: An Evolving Methodology for Flight Control Design. In *AGARD Conference Proceedings 560: Active Control Technology: Applications and Lessons Learned*, pages 7–1 – 7–12, Turin, Italy, May 1994. NATO-AGARD.

[10] European Aviation Safety Agency (EASA). Certification Specifications for All Weather Operations CS-AWO, 2003. Available from `http://www.easa.europa.eu/home/certspecs_en.html`.

[11] Christopher Fielding, Andras Varga, Samir Bennani, and Michiel Selier (Eds.). *Advanced Techniques for Clearance of Flight Control Laws*. Lecture Notes in Control and Information Sciences 283. Springer Verlag, London, 2002.

[12] H. Friehmelt and P. Huber. Vector- Die X31A fliegt zu neuen bahnbrechenden Technologiedemonstrationen. In *DGLR Luft- und Raumfahrtkongress 2001, Hamburg 17.-20. Sept. 2001*.

[13] 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.

[14] F. G. Lemoine, S. C. Kenyon, J. K. Factor, R.G. Trimmer, N. K. Pavlis, D. S. Chinn, C. M. Cox, S. M. Klosko, S. B. Luthcke, M. H. Torrence, Y. M. Wang, R. G. Williamson, E. C. Pavlis, R. H. Rapp, and T. R. Olson. The Development of the Joint NASA GSFC and NIMA Geopotential Model EGM96. Technical Report NASA/TP-1998-206861, NASA Goddard Space Flight Center, Greenbelt, Maryland, July 1998.

[15] Gertjan Looye. Integrated Flight Mechanics and Aeroelastic Aircraft Modeling using Object-Oriented Modeling Techniques. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, Portland, USA, August 1999. AIAA-99-4192.

[16] Gertjan Looye. Design of Robust Autopilot Control Laws with Nonlinear Dynamic Inversion. *at – Automatisierungstechnik*, 49(12), 2001.

[17] Gertjan Looye. *Rapid prototyping using inversion-based control and object-oriented modelling.*, chapter 8. Lecture Notes in Control and Information Sciences. Springer Verlag, Berlin, 2007.

[18] Gertjan Looye, Michael Thümmel, Matthias Kurze, Martin Otter, and Johann Bals. Nonlinear Inverse Models for Control. In *Proceedings of the third international Modelica conference*, Hamburg, March 2005.

[19] 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`.

[20] Jean-François Magni, Samir Bennani, and Jan Terlouw (Eds). *Robust Flight Control – A Design Challenge*. Lecture Notes in Control and Information Sciences 224. Springer Verlag, London, 1997.

[21] The Math Works Inc. *MATLAB – External Interfaces Reference – Version 7*, March 2005.

[22] The Math Works Inc. *Simulink Reference*, 2005.

[23] Susan McLean, Susan Macmillan, Stefan Maus, Vincent Lesur, Alan Thomson, and David Dater. The US/UK World Magnetic Model for 2005-2010. Technical Report NOAA Technical Report NESDIS/NGDC-1, NOAA National Geophysical Data Center, British Geological Survey Geomagnetism Group, December 2004.

[24] D. Moormann and G. Looye. The modelica flight dynamics library. In *Proceedings of the 2nd International Modelica Conference*, pages 275–284. The Modelica Association, 2002. Available from `http://www.modelica.org`.

[25] D. Moormann, P.J. Mosterman, and G. Looye. Object-oriented computational model building of aircraft flight dynamics and systems. *Aerospace Science and Technology*, 3(3), April 1999.

[26] Dieter Moormann. Physical modeling of controlled aircraft. In *Proceedings of the CESA'96 IMACS Multiconference on Computational Engineering in Systems Applications*, pages 970–975, Lille-France, July 1996.

[27] Dieter Moormann. *Automatisierte Modellbildung der Flugsystemdynamik*. VDI Verlag, Reihe 8, Dissertation RWTH Aachen, Institut für Flugdynamik, Düsseldorf, 2002.

[28] Martin Otter, Hilding Elmqvist, and Sven Erik Mattsson. The new modelica multibody library. In Peter Fritzson, editor, *Proceedings of the 3^{rd} International Modelica Conference, Linköping, Sweden*, pages 311 – 330. Modelica Association, 2003.

[29] S.J. Rasmussen and S.G. Breslin. AVDS: a Flight Systems Design Tool for Visualization and Engineer-in-the-Loop Simulation. AIAA 97-3467. 1997.

[30] Christian Reschke. Flight loads analysis with inertially coupled equations of motion. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2005. AIAA 2005-6026.

[31] W.F.J.A. Rouwhorst. Robust and Efficient Autopilots control Laws design, demonstrating the use of modern robust control design methodologies in the autoland system design process – the REAL Project. In *Proceedings of the Aeronautics Days 2001, Hamburg Germany*, January 2001.

[32] R. Steinhauser, G. Looye, and O. Brieger. Design and Evaluation of Control Laws for the X-31A with Reduced Vertical Tail. In *Proceedings of the AIAA Guidance and Control Conference*, Providence, Rhode Island, USA, August 2004. AIAA-2004-5031.

[33] Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation*. Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 2000.

[34] Martin R. Waszak and Dave K. Schmidt. Flight Dynamics of Aeroelastic Vehicles. *AIAA Journal of Aircraft*, 25(6):563–571, June 1988.