IFASD 2015
International Forum on Aeroelasticity and Structural Dynamics
June 28 — July 02, 2015 Saint Petersburg, Russia

EMBEDDING A MULTI-BODY MODEL OF A FIXED-WING AIRCRAFT IN A NONLINEAR STATE OBSERVER

Tuur L. Benoit[1], Yves C. Lemmens[1], Wim A. Desmet[2]

[1] Siemens PLM Software, Simulation and Test Solutions
Interleuvenlaan 68, 3001 Leuven, Belgium
tuur.benoit@siemens.com

[2] KU Leuven, PMA
Celestijnenlaan 300, 3001 Leuven, Belgium

Keywords: health monitoring, state estimation, observer, loads, aeroelasticity.

Abstract: This paper proposes a solution for utilizing multi-body models in nonlinear state observers, to directly estimate the loads acting on the aircraft structure from measurement data of sensors that are commonly available on modern aircraft, such as accelerometers on the wing, rate gyros and strain gages.

A high-fidelity aeroelastic multi-body model of a fixed-wing large passenger aircraft is presented, suitable for the monitoring of landing maneuvers. The model contains a modally reduced flexible airframe and aerodynamic forces modeled with a doublet-lattice method. In addition, detailed multi-body models of the nose and main landing gear are attached to the flexible structure, allowing to accurately capture the loads during a hard landing event.

It is expected that this approach will make way for embedding non-linear multi-body models, with a high number of degrees of freedom, in state estimation algorithms, and hence improve health monitoring applications.

1 INTRODUCTION

In the last decades, aircraft have evolved from pure mechanical to complex mechatronic systems. Avionics, actuators, sensors and the Fly-by-Wire (FBW) control system which drives these systems have become standard technology and fulfill critical functions in the safety of the aircraft [1].

Modern aircraft have many sensors onboard that are used for both control of the aircraft and monitoring the aircraft’s condition. For health monitoring applications, the sensor data is logged to the flight data recorder. On the ground, this data is downloaded from the flight data recorder and a criticality assessment process is performed in order to detect if the loads on the aircraft’s structure have exceeded the design loads during flight or landing.

For legacy aircraft the criticality is assigned based on a look-up chart where the functions are based on key parameters such as vertical and lateral acceleration, aircraft mass, roll, pitch, yaw rate. For more recent aircraft, neural network algorithms are employed to assign the criticality based on the same key parameters. The look-up charts and neural networks are generated based on a large number of loads simulations and stress analysis [2].

However, there is a need for improvement of this process. The correctness of the assignment depends on the accuracy of the look-up charts or neural networks. Since the conditions of the real event will differ in general from the simulated conditions, it is possible that false positives occur or that potential overloads remain undetected. This approach is limited in its capability of classifying events, because the assignment is black-box algorithm based on external key parameters such as acceleration, angular rates and mass and not on the loads itself .

The reason that the detection nowadays does not take into account the loads is because it is not practically feasible to measure them on an operational aircraft. This paper presents a method for observing the states of an aeroelastic multi-body model based on measurement data of sensors that are commonly available on modern aircraft and an aeroelastic model of the aircraft. The loads that act on the aircraft can then be derived directly from the states of the model.

Estimation algorithms such as the Kalman filter and its many variants [3]–[5] have long been used to estimate, from the sensor data, quantities that are important for the control system such as airspeed, vertical speed, angle of attack and sideslip angle. However, the models that are used today in the estimators are often linear models with few degrees of freedom [6], [7]. For many applications, such as control, this may be sufficient, however health monitoring applications could profit from the use of a high-fidelity non-linear model with many degrees of freedom in the state observer.

The benefits of including a high-fidelity model in the state estimation algorithm are:

1. More accurate estimates could be provided, because more physical phenomena can be caught with the high-fidelity model.
2. Quantities that are difficult to measure in practice can be observed through ‘virtual sensing’ on the model.
3. More sensor data can be incorporated in the estimator. ‘Sensor fusion’ is the science of combining multiple sources of information for producing more accurate estimates

Often, such high-fidelity models are available in the aeronautics industry in a systems engineering context but nowadays few of these models are reused in an operational context due to the challenges that are faced: in design, the models are used for simulation only in a closed software environment, while for health monitoring applications these models have to be combined with sensor data. This requires that the software has an open interface to handle the data coming from the sensors.

This paper proposes a solution for utilizing multi-body models in nonlinear state estimators. A commercial-off-the-shelf tool for multi-body simulation is embedded in the state estimation algorithm via the FMI 2.0 standard for co-simulation and model exchange [8]. This interface enables the coupling of two or more simulation tools by providing a means to exchange information regarding the integration of the continuous-time dynamical equations but is also beneficial for embedding the model in a state estimation algorithm.

The paper is organized as follows: the first three sections introduce the different parts of the estimation algorithm. The first section introduces the multi-body equations. The extended Kalman filter is introduced in the second section. Section 3 discusses how the FMI interface can be used for coupling the multi-body model and the state estimator. Section 4 brings the pieces together and shows the complete algorithm. Finally an aeroelastic multi-body model of a large passenger aircraft is presented that can be used for state estimation.

2 MULTI-BODY

Multibody simulation allows an engineer to create virtual entities (bodies) with the properties of a designed component and connect these together through joints or constraints in order to mimic the behavior of the full mechanism. The dynamics of such systems are often very complex and are governed by complex relationships resulting from the relative motion of the different bodies of the mechanism and the forces acting between them.

In this section the kinematic and dynamic equations of a system of interconnected bodies are introduced briefly. This will result in a system of Differential Algebraic Equations (DAEs). Furthermore, this section will also present the equations in a form that is suitable for embedding the system in Kalman filters.

Different choices are possible for the coordinates in which the system is described. Each type of coordinates will lead to a different formulation of the equations of motion. The most used approaches are listed below.

1. Relative coordinates: in this approach the position of a body is described relative to the previous body. This description leads to a system of Ordinary Differential Equations (ODEs) with a minimal number of degrees of freedom.
2. Natural coordinates: In this formulation, the connection points on a body are used as coordinates and the connections are described by constraint equations. Due to the nature of the formulation, these constraint equations are often very simple (quadratic).
3. Cartesian coordinates: this is the most widely used representation used in commercial software. The coordinates are described as the location of the center of gravity of a body in a global axis system. This formulation has an intuitive interpretation and is easily extended to flexible multi-body simulation. However it leads to a set of highly nonlinear DAEs.

In the remainder of this section, the equations of a multi-body system described with set of redundant Cartesian coordinates are presented without any further derivation. The interested reader is referred to [9] for a formal derivation of the equations. This formulation, also known as the augmented formulation, is used by LMS Virtual.Lab™ Motion software, the simulation software used for the model described in this paper.

The vector of generalized coordinates, containing both the independent and the dependent coordinates, is written as

$$q = [q_1 \quad q_2 \quad \cdots \quad q_n]^{\mathrm{T}} \tag{1}$$
$$q_i = [R_x \quad R_y \quad R_z \quad \theta_0 \quad \theta_x \quad \theta_y \quad \theta_z]$$

where R_i and θ_i are the absolute Cartesian coordinates and the orientation coordinates of body i. The orientation is represented with a set of four redundant Euler parameters, to avoid singularities which occur when using e.g. Euler angles.

The kinematic relationships are written as a general nonlinear algebraic equation:

$$C(q,t) = 0 \tag{2}$$

where t represents time. The velocity kinematic equations are obtained by differentiating Eq. 2 with respect to time:

$$C_q \dot{q} = -C_t \tag{3}$$

where $C_q = \partial C/\partial q$ and $C_t = \partial C/\partial t$. Differentiating Eq. 3 with respect to time gives the kinematic acceleration equations:

$$C_q\ddot{q} = Q_d \tag{4}$$

where $Q_d = -C_{tt} - \left(C_q\dot{q}\right)_q\dot{q} - 2C_{qt}\dot{q}$ gathers all the velocity dependent terms.

The dynamic equations are formulated in terms of a set of dependent and independent coordinates q. The kinematic relationships which describe the joints are added to the system by means of Lagrange multipliers.

$$\begin{bmatrix} M & C_q^{\mathrm{T}} \\ C_q & 0 \end{bmatrix}\begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q_e + Q_v \\ Q_d \end{bmatrix} \tag{5}$$

In this equation, M is the mass matrix, λ is the vector of Langrange multipliers, Q_e is the vector of generalized applied forces, and Q_v is the vector of inertia forces, that includes velocity dependent terms. Note that, although this system of equations is written in matrix form, it is not a linear system, since the matrix elements are function of the coordinates q itself. Furthermore, by inverting the left-hand side matrix, Eq. 5 can also be written in a general form

$$f(q, \dot{q}, t) = \ 0 \tag{6}$$

To summarize, Eq. 2 together with Eq. 6 describe the kinematics and dynamics of a multi-body system.

3 KALMAN FILTER

A Kalman filter is an algorithm which uses noisy measurements and a model of the system to produce an estimate of a variable that tends to be more precise than an estimate based on measurements only. The original algorithm [3] was proposed for linear time-invariant models. The discrete extended Kalman filter is a variant of the filter, which allows the use of nonlinear models, by linearizing about the current estimate [9].

The filter is presented here in discrete form since that is how the model looks like from the point of view of the estimator when communicating through the FMI interface for co-simulation (discussed in the next section).

The discrete extended Kalman filter is usually split into two steps.

1. In the prediction step, the model equations and estimation-error covariance are integrated in time, obtaining the a-priori estimate $\hat{x}_k^-$ and error covariance P_k^-.

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T \tag{7}$$

$$\hat{x}_k^- = f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) \tag{8}$$

2. In the correction step, the a-priori estimate $\hat{x}_k^-$ and error covariance P_k^- are updated with measurements y_k, obtaining the a-posteriori estimate $\hat{x}_k^+$ and error covariance P_k^+.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \tag{9}$$

$$P_k^+ = (I - K_k H_k)P_k^- \tag{10}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[y_k - h_k(\hat{x}_k^-, 0)] \tag{11}$$

Where a subscript f_k denotes evaluation of f at time instant k.

K_k is called the Kalman gain and represents the relative importance of the model prediction with respect to the measurements. A filter with a high gain will follow the measurements more closely, while a low gain will give estimates that follow the model more closely.

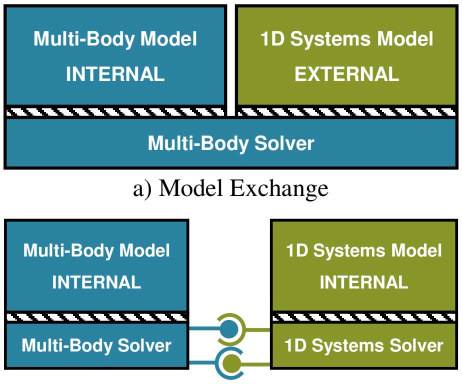
Furthermore, $F_{k-1} = \partial f_{k-1}/\partial x$ and $L_{k-1} = \partial f_{k-1}/\partial w$ are partial derivatives of the system equations with respect to respectively the states and the model noise; while $H_k = \partial h_k/\partial x$ and $M_k = \partial h_k/\partial v$ are partial derivatives of the model output with respect to respectively the states and the measurement noise. As from version 2.0 of FMI, these matrices can be exposed through the interface to the state estimator.

4 FMI 2.0

The Functional Mockup Interface (FMI) is a tool independent standard for co-simulation and model exchange [8]. It consists of a set of functions in form of a C-code file and an xml file with the description of the model structure, e.g. number and types of states, derivatives, inputs, and outputs.

Two main approaches are used to link two subsystems. Both of them are supported by the FMI interface. Figure 1 shows schematically the difference between both approaches, for a case where a multi-body model and a 1D systems model are coupled.

1. In *model exchange*, the equations of one subsystem are incorporated into the other, and only one solver integrates the overall system. The coupling is fulfilled in a *strong* sense, meaning that the two systems are coupled at each *integration time step*.
2. In *co-simulation*, each solver solves the equations of its own subsystem, and only at discrete time steps information is exchanged between the solvers. This coupling is a *weak* coupling, since the systems are only coupled at each *communication time step*.



a) Model Exchange

b) Co-Simulation

Figure 1. In model exchange, the equations of one subsystem are incorporated into the other, whereas in co-simulation each solver solves the equations of its own subsystem.

Model exchange is the preferred method when coupling two systems that have a strong interaction. The full capabilities of the solver can be exploited, such as variable step sizes and iterative methods, giving good numerical stability. However, subsystems that are differently stiff can dictate small integration steps and may therefore lead to an inefficient solution.

Co-simulation, has the advantage that problem-specific solvers can be used that are well adapted to the specific subsystem, and therefore may be more computational efficient. However, since information is only exchanged at particular time instants, co-simulation can give numerical instabilities, leading to a solution that doesn't converge. In addition, in case of co-simulation, accuracy will also decrease because of the discretization error. Extrapolation, interpolation and iterative co-simulation schemes exist that tackle these accuracy and stability issues [10].

4.1 FMI in Context of State Estimation

In a state estimation context, only one simulation tool is involved: the multi-body solver. It is coupled to the state estimator. This paper proposes the use of a discrete Kalman filter. In this case, FMI is only used for communicating certain data, e.g. the states, updated states, and Jacobian matrices at particular time instants. The state estimator is not acting as a separate system directly interacting with the multi-body mechanism, rather, it only provides updates of the states (Eq. 11). In such case co-simulation is an appropriate choice.

First, the accuracy of the solution is not affected by co-simulation since only one simulation tool is involved, hence the discretization error due to the co-simulation scheme does not appear.

Second, for the same reason, the stability of the solution is not affected by the co-simulation, rather, it is dictated by the stability of the estimation algorithm, which is a separate concern.

Third, by using co-simulation, the multi-body solver handles the time integration of the equations. The solver can therefore take full advantage of the internal representation of the equations. Since the equations are of DAE form, solving this system requires special attention, in order to make the solution consistent with the constraints. This is further discussed in the next section. The co-simulation scheme enables the use of such a problem-specific solver.

4.2 FMI 2.0 for Co-Simulation

FMI for co-simulation provides an interface for solving coupled systems that are continuous in time or time-discrete. In co-simulation, the following assumptions are made:

1. Communication between subsystems takes only place at discrete time instants, called communication points.
2. Seen from the outside, a subsystem is a sampled-data systems (e.g. controller), or a hybrid ODE that is integrated between the communication points (e.g. multi-body), or both.

Within the subsystem, events such as discontinuity handling may happen, but are not visible from outside. Likewise, systems that are governed by DAEs appear to be ODEs from the outside, but may have internal routines that make the states consistent with the constraints. Therefore, the equations of a subsystem, as seen through the FMI interface have the form of a discrete system:

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}) \tag{12}$$

$$y_k = h_k(x_k, u_k) \tag{13}$$

where f is calculates the value of the state x_k at the current time from the value of the state x_{k-1} and input u_{k-1} at the previous time step, and where h gives the output of the system y_k at the current time.

The co-simulation interface realizes two basic functionalities:

1. Synchronization of the different subsystems, and
2. Data exchange between the subsystems.

Different synchronization schemes are possible, with the ability of handling variable communication step sizes, and many are supported by the FMI standard [8], however this paper uses a standard, widely used, fixed step communication scheme, in which data is exchanged at fixed, predefined time instants.

Version 2 of FMI introduces the ability to exchange Jacobians; that is, derivatives of functions with respect to their variables. Initially it was conceived for advanced interpolation schemes, such as wave-form iteration [10], to improve the stability and accuracy of the simulation.

It is clear that this functionality can be exploited for state estimation too. In equation 7 partial derivatives of the system equation appear, and partial derivatives of the output equation appear in equations 9, 10 and 11. It is therefore desired to adopt the new functionality in the FMI 2.0 interface to obtain the value of these Jacobians directly. This avoids having to perform a tedious calculation to obtain the Jacobian by finite differencing. In addition, some solvers, such as LMS Virtual.Lab Motion, can provide Jacobians that are calculated analytically, and therefore can be more accurate, and faster to compute.

5 SIMULATION AND STATE ESTIMATION

The previous sections provided an overview of the different actors in the state estimation algorithm. Before all the pieces can be put together, some gluing is necessary. This section first shows how to convert the second order equations to first order for embedding in the Kalman filter. It continues with a discussion of the time discretization of the equations and a method for solving the constraints. Finally, it concludes with an overview of all the steps of the state estimation algorithm.

5.1 First Order Form of the Dynamic Equations

The equations governing multi-body systems have been presented in second order form (Eq. 6). However, the formulation for the Kalman filter (Eqs. 7 – 11) has been derived for first order ODEs. In order to embed the multi-body model in a state estimator, the state vector x is defined as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{14}$$

With this new definition Eq. 6 becomes:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ f(q, \dot{q}, t) \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ f(x, t) \end{bmatrix} \tag{15}$$

Eq. 15 represents the state space equations of the multi-body system and are of first order.

5.2 Discretization

Equation 15 is still a continuous-time representation of the model. On the other hand, the discrete extended Kalman filter presumes a model in discrete-time form: functions f_{k-1} and h_k in equations 8 and 11 are sampled-data functions that calculate values for discrete time instants.

The key point here, is not to discretize the model equations directly, but instead, by communicating through the FMI interface, the discretization scheme is abstracted away. So, from the point of view of the estimator, there is only a discrete system, as governed by equations 12 and 13. While, in fact, the multi-body solver internally discretizes the equations in time using a particular integration scheme.

This can be an explicit integration scheme, such as Runge-Kutta, or more preferably for stiff multi-body systems, an implicit method such as the Backward Differentiation Formula (BDF). The choice is independent from the Kalman filter algorithm and can be tuned to the specific model.

Furthermore, although the Kalman filter has a fixed step size, equal to the communication time step size, it is not obligatory to also have a fixed step integrator. It is perfectly fine that the multi-body solver takes multiple integration steps of variable size in between one communication interval in order to meet the tolerances on the integration error.

It should be noted that a variable time-step solver will eliminate the possibility of real-time state estimation, because that requires deterministic behavior. However, for health-monitoring applications, it is in most cases not required to have the results in real-time. For example, a slight delay of several minutes is acceptable for the monitoring of landing maneuvers: the time for the aircraft to reach the gate is in the same order of magnitude.

5.3 Solving the Algebraic Equations

If the equations governing multi-body system would have been ODEs, then one would have been able to perform state estimation at this moment. However, due to the augmented formulation, the state vector contains redundant coordinates, for which the values are constraint by a system of nonlinear algebraic equations (Eq. 2).

The classic way of solving such a DAE system, is to split the state vector into a set of independent and a set of dependent coordinates. The equations are then integrated in time for the independent coordinates only. Afterwards a solution is sought for the dependent coordinates.

A decomposition of the constraint Jacobian matrix C_q is performed to identify a set of independent coordinates. The selection of the independent coordinates is important because a bad choice of the independent coordinates can give numerical difficulties or lead to a singular configuration. The right set of coordinates depends on state of the mechanism and can change during the simulation. Therefore the selection is updated when the numerical error becomes excessive.

Having determined a solution for the independent coordinates, a solution for the dependent coordinates can be constructed using a Newton-Raphson algorithm. This algorithm projects the dependent coordinates on the space defined by the constraints.

The estimator only sees an ODE system, as described by equations 12 and 13. The estimator has no knowledge of the real DAE nature of the equations and thus cannot perform the Newton-Raphson algorithm for satisfying the constraints. In this case, the estimator relies on the multi-body solver to perform this calculation automatically, during the time update of the states, e.g. during the calculation of f_{k-1} in Eq. 8.

State estimators that handle constraints directly do exist and are described in literature; examples are constraint Kalman filtering [9] and moving horizon estimation [11]. However, since the constraint equations cannot be exported by the FMI interface, this type of estimators is not suitable for use with a commercial multi-body solver such as LMS Virtual.Lab Motion at this moment.

5.4 The Complete Algorithm

The complete algorithm for performing simulation and state estimation with a multi-body model is shown in Figure 2. Figure 2 also shows the interaction between the multi-body solver and the state estimator and which data is exchanged through the FMI interface. The main steps are summarized below.

1. Initial conditions define the initial configuration $\hat{x}_0^+$ of the multi-body system and should be a good approximation of the real system configuration. The uncertainty of the intial configuration is given by the initial error covariance P_0^+. A bad choice for the initial conditions or error covariance can undermine the convergence of the Kalman filter [9].
2. Perform the prediction step.
    a. Compute partial derivatives of the system equations to calculate the a-priori error covariance (Eq. 7).
    b. Identify a set of independent coordinates by performing a decomposition of the constraint Jacobian matrix C_q (in Eq. 3).
    c. The independent accelerations are integrated forward in time using a numerical integration method. The numerical solution defines the independent coordinates and velocities (Eq 5).
    d. Use the values of the independent coordinates to solve the system of algebraic constraint equations for the dependent coordinates (Eq. 2). This system can be solved iteratively with a Newton-Raphson method.
    e. Use the result to calculate the a-priori estimate (Eq. 8).
3. Perform the correction step.
    a. Compute partial derivatives of the output equations to calculate the Kalman gain (Eq. 9) and the a-posteriori error covariance (Eq. 10).
    b. Compute the model output and perform the measurement update, to calculate the a-posteriori estimate (Eq. 11).
4. Go to step 2 and repeat. This process continues until the end of the simulation time is reached.

Note that the algorithm contains multiple nested iterations. A first inner iteration, performs the time integration of the model equations between two communication time steps. The second inner iteration is an iterative method to solve the constraint equations. The outer loop increments time, establishing the time-domain simulation.

Supplying this algorithm with measurement data from the flight data recorder, will give an estimation of the states of the model, in case a set of sensors is selected that makes the system observable. Once the states of the system are known, an assessment of the observed maneuver can be performed equal to the post-processing a normal simulation. It is then possible to check the interested quantities, directly in the model, even if these quantities were not measured on the real aircraft. For example, it is then possible to obtain the bending and torsion moment at the wing root although this was not measured directly. This is called "virtual sensing".

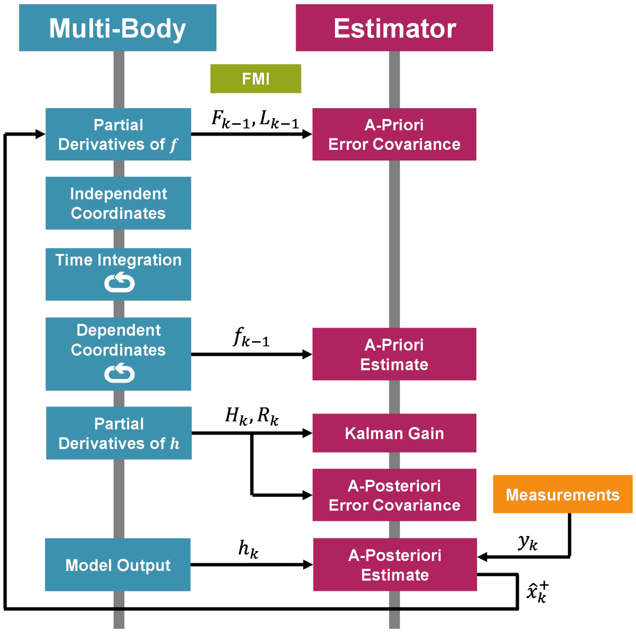


Figure 2. Algorithm for state estimation with a multi-body model.

6 MODEL

The simulation of landing maneuvers requires a careful consideration of all forces acting on the aircraft in order to capture the loads accurately. The aerodynamic forces are distributed over the complete surface of the aircraft and are highly influenced by the flexibility of the airframe. Engine forces, on the other hand, are localized forces acting on the pylons. Similarly, landing gear forces are localized, acting only at the small number of connection points with the airframe.

The landing gear forces are very nonlinear, however, due to the following reasons:

1. The landing gear is a complex mechanism consisting of a dozen of interconnected bodies. A landing gear deflection changes the geometry in such way that large rotational deviations occur, causing nonlinear behavior.
2. The shock absorber is a complex hydraulic system with very nonlinear stiffness and damping characteristics. Both stiffness and damping change very rapidly during the landing gear deflection and are different during compression or expansion.
3. The tyre is an important part of the landing gear that influences the forces. The tyre force is a highly nonlinear function of parameters such as slip angle, wheel speed, road surface, tyre wear and tear, etc.

The origin and manifestation of the different forces acting on the aircraft are quite distinct and therefore require separate treatment.

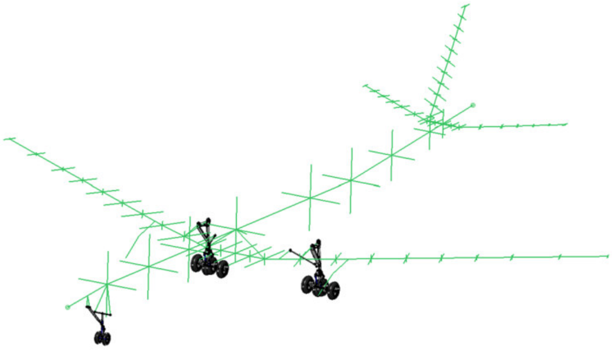


Figure 3. Complete model which integrates a flexible airframe and models for nose and main landing gear.

Figure 3 shows a complete model of a large fixed-wing passenger aircraft for the simulation of landing maneuvers. It is an extension of a model for aeroelastic analysis [12], with the addition of three landing gear. While the original aeroelastic model could be used to simulate fluid-structure interaction to calculate gust responses for load calculation as well as to predict flutter, this new model additionally allows the simulation of landing maneuvers to obtain the loads in case of high impact events such as a hard landing. The following subsections describe each part of the model in more detail.

6.1 Flexible Airframe

An aeroelastic model is a combination of a structural dynamics model and an aerodynamic model. Figure 4 shows the structural dynamics model of the airframe of the large passenger aircraft. It is a stick model, for which the structural elements are modelled as beams. The properties of the beam, e.g. stiffness in the different directions and density, are matched such that they are equivalent with the real structure.

One way of obtaining the properties is to correlate the model with experimental data from a Ground Vibration Test (GVT). During a GVT, the real aircraft is instrumented with accelerometers on the fuselage and wings, and by exciting the structure and measuring the response, the structural modes can be determined. By correlating the modes of the beam model with the experimentally obtained modes, the stiffness and mass is obtained.

Another way for obtaining the properties is by reducing a higher fidelity finite element model that starts from first principles. In this case, a detailed finite element model is created by meshing the geometry of the aircraft and assigning material properties for each part. This full 3D model is divided in sections and for each section an equivalent beam is calculated.

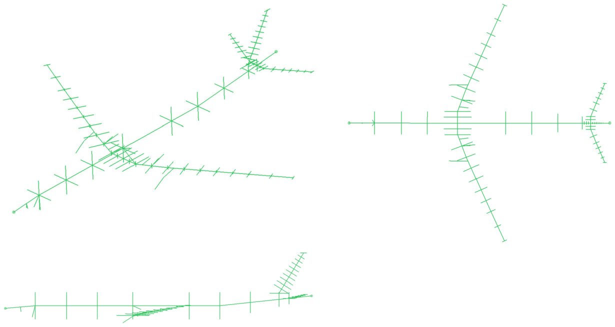


Figure 4. Stick model of the flexible airframe, for which the structural elements are represented by equivalent beams.

The number of Degrees Of Freedom (DOFs) of the stick model is still too large for embedding in multi-body simulation. To decrease the DOFs further, modal reduction is applied. A linear systems, such as a linear structural dynamics model, can be represented as:

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{C}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{f}(t) \tag{16}$$

where $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ are the mass, damping and stiffness matrices respectively. Vector $\boldsymbol{f}$ contains the external force and $\boldsymbol{x}$ is the displacement of the nodes. By the coordinate transformation

$$\boldsymbol{x} = \boldsymbol{\phi}\boldsymbol{q}\,, \tag{17}$$

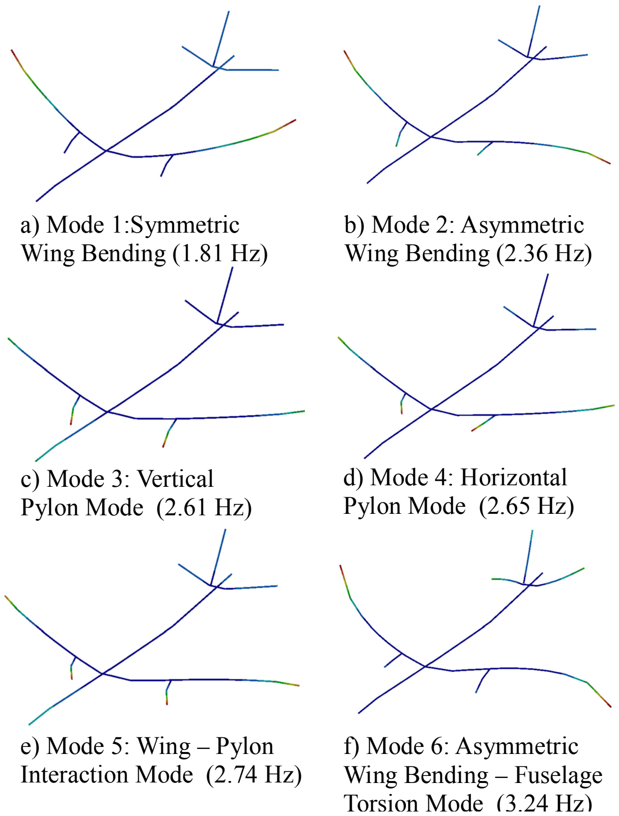
equation 17 is transformed to a modal base, producing:

$$\boldsymbol{M}_q\ddot{\boldsymbol{q}} + \boldsymbol{C}_q\dot{\boldsymbol{q}} + \boldsymbol{K}_q\boldsymbol{q} = \boldsymbol{f}_q \tag{18}$$

where $\boldsymbol{M}_q = \boldsymbol{\phi}^T\boldsymbol{M}\boldsymbol{\phi}$, $\boldsymbol{C}_q = \boldsymbol{\phi}^T\boldsymbol{C}\boldsymbol{\phi}$, $\boldsymbol{K}_q = \boldsymbol{\phi}^T\boldsymbol{K}\boldsymbol{\phi}$ are the modal mass, damping and stiffness matrices and $\boldsymbol{f}_q = \boldsymbol{\phi}^T\boldsymbol{f}$ the modal force vector.

It can be shown that the modal mass and stiffness matrices are diagonal and with proportional damping $\boldsymbol{C}_q$, all equations are fully decoupled. Finally, the reduction is achieved by keeping only a subset of the modes: the modes that contribute the most to the dynamic respons. The other modes, and the corresponding DOFs, are disregarded, resulting in a reduced system with fewer DOFs.

For the large passenger aircraft, the six rigid modes and the first 41 lowest modes are kept, spanning frequencies from 1.81Hz up to 30Hz for the highest flexible mode. Figure 5 shows the first six flexible modes. Note that a modal reduction of the structure is also convenient from an aerodynamic point of view, since the aerodynamic model can be reduced to a modal base in a similar fashion.


a) Mode 1:Symmetric Wing Bending (1.81 Hz)

b) Mode 2: Asymmetric Wing Bending (2.36 Hz)

c) Mode 3: Vertical Pylon Mode (2.61 Hz)

d) Mode 4: Horizontal Pylon Mode (2.65 Hz)

e) Mode 5: Wing – Pylon Interaction Mode (2.74 Hz)

f) Mode 6: Asymmetric Wing Bending – Fuselage Torsion Mode (3.24 Hz)

Figure 5. First six modes of the modally reduced model of the airframe.

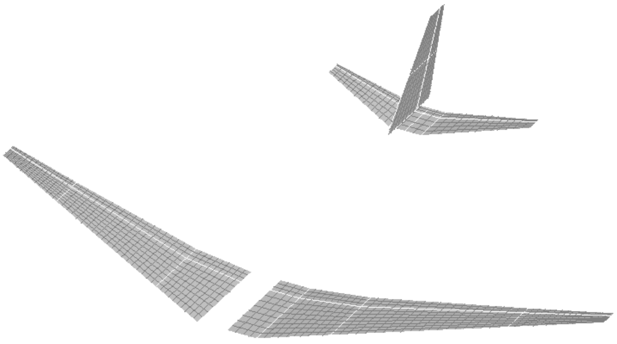


Figure 6. The aerodynamic model, showing the panels of the doublet-lattice method.

6.2 Aerodynamic Model

Panel methods, such as the doublet lattice method, are often used to represent the aerodynamic forces acting on an aircraft. Although Computation Fluid Dynamics (CFD) simulation can provide a much more detailed and accurate calculation of the aerodynamic forces, in the majority of aeroelastic calculations panel methods are used for their simplicity and fast computation. In addition, they can be combined very efficiently with a structural dynamics model.

The wing is divided into chordwise and spanwise panels, as can be seen on Figure 6. The lift on each panel can be determined as

$$\boldsymbol{L} = \frac{\rho V^2}{2} \boldsymbol{AIC\alpha} \tag{19}$$

where $\boldsymbol{L}$ is the vector of lift forces on each panel, $\boldsymbol{\alpha}$ is the vector with the angles of incidence of each panel, and $\boldsymbol{AIC}$ is the matrix of aerodynamic influence coefficients. The $\boldsymbol{AIC}$ matrix relates the lift on each panel to the aerodynamics and angle of incidence. The values of this matrix can be determined by placing a single vortex, source, sink or doublet, at the quarter chord of each panel and enforcing zero normal flow condition at the boundary of the wing surface. A detailed exposition of panel methods and determination of the coefficients is found in [13].

The lift forces in Eq. 19 are given for each physical panel, and are based on physical displacements of the structure. It is more efficient to express the forces in the modal domain, as function of modal parameters. By doing this, the aerodynamic forces are also compatible with the structural dynamics model that is formulated in the modal space. The modal force vector $\boldsymbol{Q}$ is given by pre-multiplying the lift force vector $\boldsymbol{L}$ by the transpose of the modal matrix defined at the panel grid points $\boldsymbol{\Phi}$. Note that this transformation relates aerodynamic panel grid points with modal coordinates and is therefore different from $\boldsymbol{\phi}$ in Eq. 3, which relates structural displacements to modal coordinates. The vector of changes of incidence $\boldsymbol{\alpha}$ are related to the modal displacements by post-multiplying with $\boldsymbol{\Phi}_x$. It may be seen that

$$\boldsymbol{Q} = \frac{\rho V^2}{2} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{AIC\Phi}_x \tag{20}$$

For steady flow the $\boldsymbol{AIC}$ matrix is real. The panel method can be extended to unsteady flow by allowing the aerodynamic influence coefficients to be complex and function of the reduced frequency $k = \omega b/V$. Consequently, the lift forces are function of the reduced frequency k. Since the simulation is performed in time domain, it is required to approximate the AICs, given for a range of frequencies, with the so-called ration fraction approximation [14]. The matrix $\boldsymbol{Q}(s)$, containing the aerodynamic forces, is expanded in terms of the Laplace variable $s = i\omega$, such that:

$$\boldsymbol{Q}(s) = \boldsymbol{A}_0 + \boldsymbol{A}_1 \frac{sb}{V} + \boldsymbol{A}_2 \left(\frac{sb}{V}\right)^2 + \frac{V}{b}\sum_{n=1}^{N_L} \frac{\boldsymbol{A}_{n+2}}{\left(s + \frac{V}{b}p_n\right)} \tag{21}$$

The function is represented here in the Laplace domain, but can be easily converted to time-domain. To this end, the N_L lag states $\boldsymbol{q}_{a_n}$ are introduced that represent the poles p_n.

$$\boldsymbol{Q}(t) = \frac{\rho V^2}{2}\left[\boldsymbol{A}_0\boldsymbol{q} + \boldsymbol{A}_1 \frac{b}{V}\dot{\boldsymbol{q}} + \boldsymbol{A}_2 \left(\frac{b}{V}\right)^2 \ddot{\boldsymbol{q}} + \sum_{n=1}^{N_L} \boldsymbol{A}_{n+2}\boldsymbol{q}_{a_n}\right] \tag{22}$$

N_L lag terms are introduced for each modal force, hence in practice the number of lag terms should be kept limited for an efficient simulation. It has been shown that for this model adding 5 lag terms provides sufficient accuracy, while at the same time also has a good computational performance [15].

The doublet lattice method is a based on potential theory and cannot predict viscous drag, transonic effects or the turbulent flow around e.g. the flaps and spoilers. Therefore, standard industrial approach is to correct the values obtained with the doublet lattice method with experimental data from wind tunnel tests and flight tests.

6.3 Landing Gear

The main landing gear is modelled after a typical landing gear mechanism for a large passenger aircraft. It consists of multiple submechanisms which each have a distinguished function. Figure 7 show the main landing gear and its main components.

- The shock absorber is a pneumatic air-oil hydraulic system which cushions the impact of a landing and dampens the vertical oscillations. The shock absorber consists of a sliding tube that can move axially in the main fitting. The air-oil hydraulic system is modelled with a spring-damper with non-linear characteristics. This simplified force curve approach does not take into account the dynamics of the hydraulic system. In order to obtain a higher fidelity model, the multi-body should be coupled, through co-simulation, with a systems model in a 1D systems tool like LMS Imagine.Lab AMESim. In addition, bearing friction forces are modelled with an analytical expression.
- The side stay connects the main fitting with a third point on the airframe. A side stay is commonly found on larger aircraft and provides additional stiffness to the landing gear mechanism. This is also beneficial for avoiding shimmy phenomena. The side stay is modelled as a connection of multiple rigid links.
- The torsion links connect the main fitting with the bogie. Its function is to orient the bogie, which can rotate freely around the main fitting axis, forward. The two links are connected to the main fitting and bogie with revolute joints.

- The pitch trimmer mechanism causes a relative angle between the sliding tube and the bogie. Its function is to orient the landing gear wheels to a desired position for touchdown or for stowing. In addition it dampens out the vibration of the bogie around its rotation axis during taxiing or during extension and retraction of the landing gear. The orientation is achieved by a hydraulic actuator, in this model simplified by a nonlinear spring.
- The bogie is the lower part of the landing gear to which the wheels are attached. It can rotate but the rotation is constraint by the pitch trimmer mechanism. Friction force is modelled with an analytical expression.
- The wheels are modelled as separate bodies, attached to the bogie with a revolute joint. A tire model is defined for each wheel, tire deflection is calculated based on a nonlinear vertical stiffness and friction forces are function of the wheel slip.

The main landing gear is connected to the airframe at three points. No connection points were foreseen in the original airframe model. Therefore nodes for the three connections were added and attached rigidly to an existing node. One of the beam sections needed to be split into two equivalent beam sections in order to provide a connection point for the side stay.

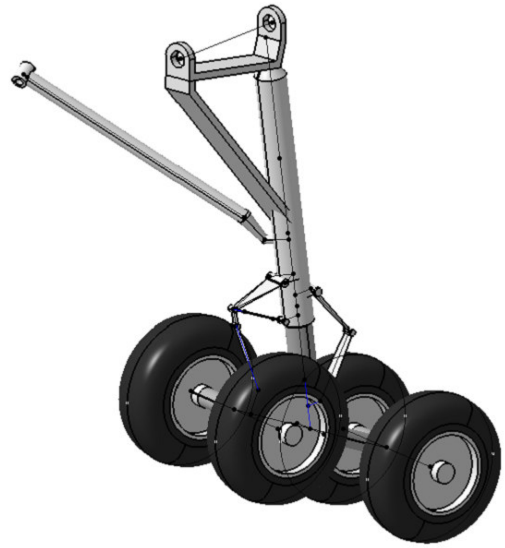


Figure 7. Multi-body model of the left main landing gear.

The nose landing gear shown on Figure 8 has many similarities with the main landing gear.

- The shock absorber is modelled with a nonlinear spring and damper, a simplified representation of the air-hydraulic system.
- Torsion links connect the main fitting and the sliding tube.
- The wheels are attached directly to the sliding tube. No bogie and pitch trimmer are present.
- A drag stay connects the main fitting with two additional points on the fuselage, giving a total of four connection points. The fore and aft connection points are connected rigidly to two nodes of the fuselage beam structure.

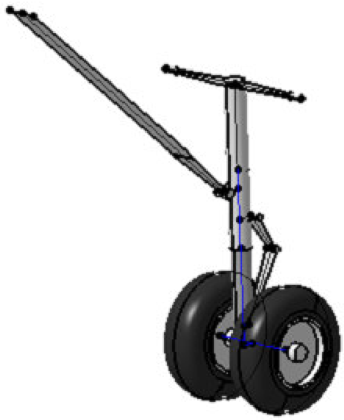


Figure 8. Multi-body model of the nose landing gear.

6.4 Model Inputs

Several inputs have an impact on the dynamics of the aircraft. First, a deflection of the control surfaces changes the lift distribution on the wing and tail, resulting in a change of the forces and moments on the structure. In this model, there are four control surfaces that can move independently: ailerons, elevator, rudder and flaps. These control surfaces are modelled in the aerodynamic model.

Second, the velocity on each aerodynamic panel is an extra input. In this model a gust shape is assumed, therefore the inputs are not independent. The one-minus-cosine gust is a idealization of a longitudinal or lateral gust in time-domain [16]:

$$U = \frac{1}{2} U_0 \left(1 - \cos\frac{2\pi x}{2H}\right) \tag{23}$$

where U_0 is the amplitude of the gust, and H the gradient distance, and x the coordinate along the gust. With a known value for the amplitude and gust length, the velocity on each panel can be determined by substituting x in Eq. 23 by the distance of the panel.

Third, engine forces are applied as a vector force at the tip of the pylon. They scale with the thrust setting, which is the input controlling the amount of thrust that the engines produce.

Finally, the nose landing gear is steerable. The nose landing gear is effective when the aircraft is on the ground, driving at low speeds. At higher speeds, the rudder is more effective and the preferred input for lateral control.

This set of inputs provides the ability to maneuver the aircraft both in the air and on the ground. It is useful for simulation of the aircraft in free-flight but it is most powerful for simulation of landing, take-off and taxi manoeuvers.

6.5 Model Outputs

The outputs of the model are chosen freely and can be almost any engineering parameter available in the model. For health monitoring application, the outputs are variables that can be measured when the aircraft is in operation. Typical sensors that are available on a modern commercial aircraft are:

- Radar and pressure altimeters that measure the location and height of the aircraft.
- Pitot tube that measures the free stream velocity of the aircraft.
- Angle-of-attack vanes that measure the angle-of-attack and side slip angles.
- Accelerometers that measure the rigid acceleration of the aircraft.
- Gyroscopes that measure the orientation and angular speed of the aircraft.
- Accelerometers and the wing tips that measure the response of the aircraft flexible structure.

For each of these sensors, it is possible to define an output in the model.

6.6 Model Order

The main landing gear each consist of 19 interconnected rigid bodies. The nose landing gear consists of 9 rigid bodies. The airframe is one flexible body with 41 modes. A rigid body has 7 DOFs as stated in Eq. 1. The number of rigid DOFs is therefore (19 * 2 + 9 + 1) * 7 = 336. With 41 modes for the flexible airframe, this adds up to 377 structural DOFs of the complete model.

The aerodynamic model uses 5 lag states per generalized force. There are 47 generalized forces to represent the aerodynamics, hence the number of DOFs increases with 47 * 5 = 235, resulting in a total model order of 612 DOFs.

7 SUMMARY

A multi-body model of a large passenger aircraft for embedding in a state estimator has been presented. It includes the structural dynamics and unsteady aerodynamics of the flexible airframe as well as a detailed model of the landing gear. The model is very well suited for the simulation of landing, take-off and ground maneuvers of aircraft subjected to wind gusts.

The use of the FMI 2.0 interface allows exchanging the information needed for coupling this model to a state estimator. Co-simulation is an appropriate choice for the coupling, because, from the point of view of the state estimator, the model is in discrete-time form which fits the formulation of the discrete extended Kalman filter well. The constraints originating from the augmented multi-body formulation are solved implicitly at each time update by the multi-body solver.

In future work, the methodology presented here will be employed to estimate the states of a flexible aircraft. The estimated states will in first instance be the modal participation of the first flexible modes, since these modes have a major contribution to the internal loads. Furthemore, the estimation of unknown distributed input forces, such as unknown gust loads, will be investigated using joint input-state estimation techniques.

8 REFERENCES

[1] D. Niedermeier and A. A. Lambregts, “Fly-By-Wire Augmented Manual Control - Basic Design Considerations,” in *International Congress of the Aeronautical Sciences*, 2012, vol. 100, p. 7.

[2] T. Wilson, “Application of Uncertainty Management to Improve Reliability of Hard Landing Criticality Assessment Criteria,” in *International Forum on Aeroelasticity and Structural Dynamics*, 2013.

[3] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[4] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nonlinear systems,” in *AeroSense’97*, 1997, pp. 182–193.

[5] S. J. Julier and J. K. Uhlmann, “Unscented Filtering and Nonlinear Estimation,” *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[6] P. Goupil, “AIRBUS state of the art and practices on FDI and FTC in flight control system,” *Control Eng. Pract.*, vol. 19, no. 6, pp. 524–539, 2011.

[7] C. D. Regan and C. V Jutte, “Survey of Applications of Active Control Technology for Gust Alleviation and New Challenges for Lighter-weight Aircraft,” 2012.

[8] FMI, “FMI for ModelExchange and CoSimulation v2.0,” 2014.

[9] D. Simon, *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*. Hoboken, N.J.: Wiley-Interscience, 2006.

[10] M. Busch and B. Schweizer, “Numerical Stability and Accuracy of Different Co-Simulation Techniques: Analytical Investigations Based on a 2-DOF Test Model,” in *Joint International Conference on Multibody System Dynamics*, 2010.

[11] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[12] K. H.H and C. J.E., “Rapid Prediction of Worst Case Gust Loads Following Structural Modification,” *AIAA J.*, vol. 52, no. 2, pp. 242–254, 2014.

[13] J. Katz and A. Plotkin, *Low speed aerodynamics*. Cambridge, UK; New York: Cambridge University Press, 2001.

[14] K. L. Roger, “Airplane math modelling and active aeroelastic control design,” *AGARD-CP-228*, pp. 4.1–4.11, 1977.

[15] A. Castrichini, V. Siddaramaiah, D. Calderon, T. Wilson, and Y. Lemmens, “Preliminary Investigation of Use of Flexible Folding Wing-Tips for Static and Dynamic Loads Alleviation,” in *4th Aircraft Structural Design Conference, Belfast*, 2014.

[16] F. M. Hoblit, *Gust Loads on Aircraft: Concepts & Applications*. Washington, D.C: AIAA, 2001.

9 ACKNOWLEDGMENTS

The research of T. Benoit is funded by a grant from the IWT Flanders.

10 ABBREVIATIONS

API	Application Program Interface
AIC	Aerodynamic Influence Coefficient
BDF	Backward Differentiation Formula
CFD	Computation Fluid Dynamics
DAE	Differential Algebraic Equation
DOF	Degree Of Freedom
FMI	Functional Mockup Interface
GVT	Ground Vibration Test
ODE	Ordinary Differential Equation
XML	Extensible Markup Language

11 COPYRIGHT STATEMENT