

Exact Method for Coupled Non-linear State-space Simulation and its Application to a Flexible Multibody Wind Turbine Aeroelastic Code

(Received: Apr. 4 2014. Revised: May 1, 2014. Accepted: May 5, 2014)

JAUME BETRAN¹
ROELAND DE BREUKER²

Abstract

During the last few decades engineers have been designing wind turbines of increasing size seeking lower values of cost of energy (CoE). A great amount of research has been done recently seeking alleviation of aerodynamic loads on the rotor and support structure and thus reducing material costs and CoE. At the same time, Finite Element Method (FEM) tools have been introduced more and more as part of the design process for structures, drive system and mechanisms of these machines. Coupling FE models with in-house modelling tools is a powerful means to enhance designs despite risking computation trouble. The present work explains a computation method to couple a comprehensive rotor aerodynamic model, including inflow and loads unsteadiness and dynamic stall, to a finite element based Multibody model of a wind turbine. All these unsteady aerodynamic phenomena behave ultimately as a set of interacting dynamic systems and are included in a proprietary in-house model of rotor aerodynamics. The methodology presented here is based on an exact closed form solution of an Ordinary Differential Equation (ODE) system with a linear-with-time source term and it aims to provide a robust and systematic way to couple in-house models in discretized time integration schemes. The presented method minimises computation trouble avoiding nested iterative procedures while getting exact results and therefore resulting in limited CPU-time and unharmed numerical behaviour. The method is validated in simple ODE cases and is finally used to build an holistic 110 m diameter wind turbine model with a view to compute loads.

1 Introduction

As modern industry seeks greater challenges while keeping costs low and weight light, complex simulations are more often included in their design methods. For bigger devices, simulation models comprise deformable parts of non-linear materials, complex topology of structures and joints, controlled actuators and loads of different nature. Examples of that are satellites, robots, civil structures, helicopters and wind turbines.

Wind energy is a good example of industry sector that has grown fast during the last three decades. After the oil crisis in the seventies, some research programmes of diverse origin and outcome were started in Europe and the US. The sector started its industrialisation in the late seventies and eighties and soon scaled the initial 50 kW, ϕ 15 m rotors to 600 kW, ϕ 50 m in the nineties, 3 MW, ϕ 100 m in the 2000s and currently reaching 6 MW, ϕ 125 m. On its way, some technology breakthroughs broke into, i.e. variable speed generators, pitch regulation, direct drive turbines, offshore foundations and, more recently, individual pitch control. At the present, an effort is being put in floating wind turbines, advanced controls and smart rotors.

Technology advances have been pushed by the growth of the market and vice-versa. In that sense, even though Europe's figures in innovation, sales and installed power are still in the top, US and Asia are very close in installed

¹ Alstom Wind

Roc Boronat, 78. Barcelona

² Faculty of Aerospace Engineering, Delft University of Technology
Kluyverweg, 1. Delft

power; China in particular has four manufacturers among the top ten. Globally speaking, the installed power quickly increased from the 10 GW during the nineties to 100 GW during the 2000s and 283 GW at end of 2012. Wind energy has in that way settled as a mainstream power generation technology worldwide. Thus, wind power share of the total electricity consumption in 2012 was 7.0% (231 TWh) in Europe, 3.5% (140 TWh) in the US, and 2% (100 TWh) in China [3, 13].

The latter scaling figures have had their mirror in the technology: whereas in the eighties and nineties wind industry had around 100 granted patents per year, some hundreds per year were issued during the 2000s and around 1500 yearly during the last few years. Most of them on Horizontal Axis Wind Turbines (HAWT).

This continuous growth is due to a big innovation effort which could not be fully explained without the assimilation of new technologies as well as modelling and computation techniques from other sectors.

Assessing durability is a key link in the design process of wind turbines, fatigue becoming a design driver for many of their components. The computation of loads in big, light elastic structures require the use of non-linear finite element tools, and modelling mechanisms must be approached with multibody tools that allow great rotations and account for all the phenomena involved in them. Practical solutions of such problem are given by time-domain simulation codes. Historically, the Wind Energy community have been using Multibody System (MBS) tools; since 2005, however, it has gradually assimilated FEM methods for components [26] and in 2012 the standards openly consider the use of FE based Flexible Multibody tools [27]; both methods often using implicit time integration schemes like the Newmark method [43].

In other sectors, in order to customise models to meet the industry needs, in-house models are sometimes used together with time integration tools to model complex problems using some kind of tight or loose coupling strategy. To guarantee a correct solution of the whole problem, a coupling method should seek the dynamic equilibrium of whole by transferring information among its parts at some level. When coupling the dynamic system with in-house models, attention should be paid in maintaining standards of numerical steadiness and computation time; even precision can be compromised in case of numerical trouble.

Frequently, in-house models are formulated as or include dynamic systems, ordinary differential equation or state-space systems. Similar problems are solved, according to the literature, using a numerical approximation to the Duhamel Integral [23, 24] or, when possible, appending the ODEs to the structural equations [9] or solving nested ODE systems using iterative algorithms. However, keeping computational time low, finding the exact solution or skipping possible unsteadiness are sometimes desirable.

In the present work a formulation of an exact closed form solution for this type of systems when used within a time integration scheme is proposed. The said formulation is used in the computation of two important phenomena involved in the unsteady aerodynamic loads of a rotor, namely, a dynamic system representing inflow velocities approaching the rotor and state-space systems representing the unsteady loads at aerofoil level at several blades stations (see figure 1).

The rest of the paper is organised as follows. Section 2 explains the state of the art of aerodynamic modelling of both aerofoil unsteady loading and rotor inflow in FE based time integration schemes and coupling simulation methods in three different subsections. Aim of the present work is described in Section 3. Section 4 explains the progress beyond the state of the art where the mathematical procedure to reach the closed-form solution is described and a verification with a simple case is given. Section 5 shows the application of this methodology to the computation of unsteady aerodynamic loads on an aerofoil and to the

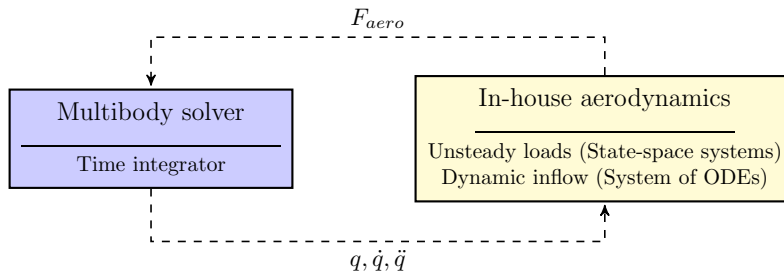


Figure 1: In-house loads code coupled with FEM based Flexible Multibody system

unsteady rotor inflow. Finally, conclusions are given in Section 6.

2 State of the Art

The present work stands on three legs, the states of the art of which are detailed below in three subsections. The first subsection is dedicated to time integration methods used in Multibody Systems. The second subsection explains numerical models of unsteady aerodynamics; two methods used to solve the Beddoes-Leishman unsteady aerodynamic loads are discussed. The third subsection reviews coupled simulation technologies used both to incorporate equations of a secondary model to an existing model in tight coupling simulation and to use two or more independent solvers in a co-simulation scheme.

2.1 Time Integration of FEM Based Flexible Multibody Problems

Holistic wind turbine simulations have been historically tackled by means of Multibody systems with rigid bodies or linear flexibility based on modal reduction techniques [6, 31, 40, 18]. More recently, catching-up with the rotorcraft industry state of the art [4, 15], and together with recommendations from the certification institutions [27], FEM based flexible multibody codes have appeared [22, 14]. Most of these comprehensive tools use implicit solvers with one-step recursive time integrators of the Newmark family or equivalent and a Newton-Raphson algorithm to reach convergence at equilibrium [43].

The flexible multibody problem is expressed as a Differential-Algebraic system of Equations (DAE):

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{J}^T \lambda = \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}, t) \\ \Phi(\mathbf{q}, t) = 0 \end{cases} \quad (1)$$

The first set of equations represents the dynamic equilibrium of all the bodies including external, internal and inertia forces, as well as constraint forces; the second set of equations represents the kinematic constraints Φ imposed by the joints; \mathbf{M} is the mass matrix, \mathbf{q} is the coordinates vector, \mathbf{g} is the force summation vector, \mathbf{J} is the Jacobian of constraints and λ is the set of Lagrange multipliers that express a measure of the internal forces needed to satisfy the corresponding set of constraints.

A Flexible Multibody problem is expressed in two sets of equations. A set of differential equations of a Flexible Multibody system represents the dynamic equilibrium of the masses and inertial and internal forces and an algebraic set of equations represents the kinematic constraints Φ imposed by the joints. The whole set of equations is solved to find dynamic equilibrium at every new time-step [12]. Firstly, a new initial prediction is proposed by the Newmark formulae; secondly, residues of coordinates and constraints are minimised with user prescribed accuracy through an iterative Newton-Raphson algorithm (see figure 2).

The Newmark implicit methods are one-step time integrators that, first predict new displacements, velocities and accelerations using data from previous

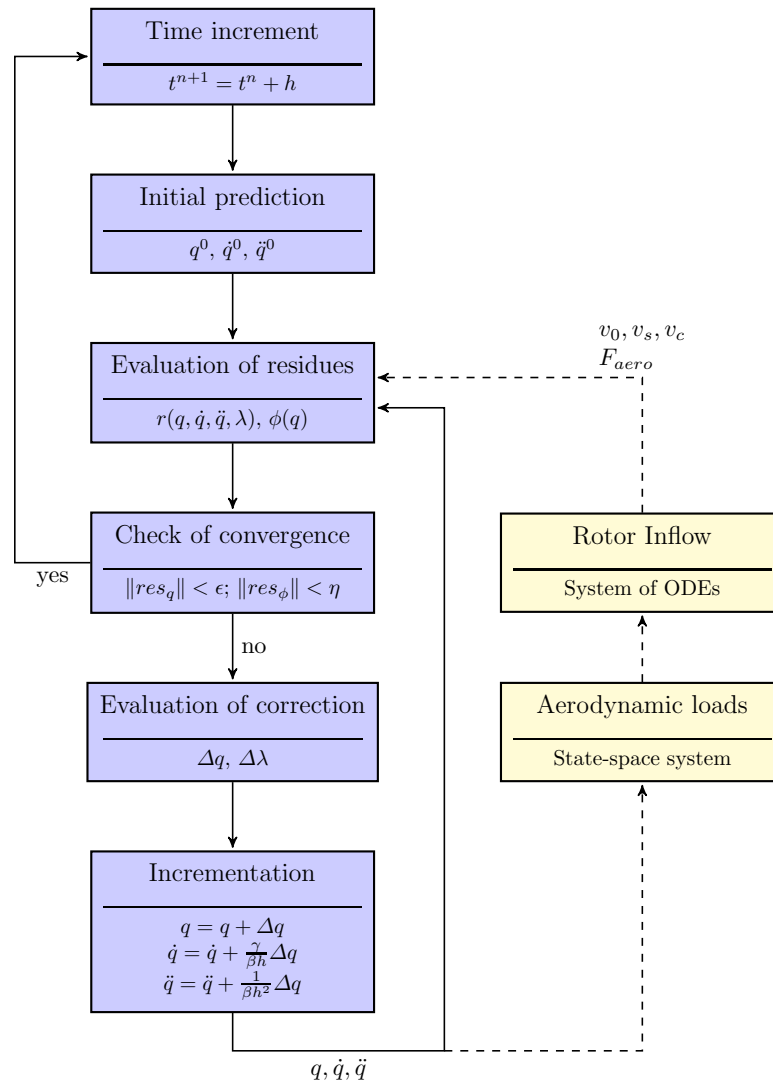


Figure 2: Newmark time integration scheme including Lagrange multipliers λ and in-house unsteady aerodynamic model with Inflow and aerofoil loads. Based on a scheme by Gradin and Cardona [12] and adapted by the authors.

time-step and, second, compute internal forces of all the elements present in the system. Residues vector is computed out of internal forces and a decision is taken on whether to iterate or not. The unsteady aerodynamic model is included as an element, therefore using previous time-step displacements (and derivatives) as inputs and passing aerodynamic loads as outputs in the form of internal loads.

2.2 Current Methods to Solve Unsteady Aerodynamic Models

A set of very established dynamic inflow models have their origin and are widely used in the modelling of helicopter flight [10]. Some of these models consider the incoming wind as a parametrised non-uniform field that responds dynamically to changes in the rotor [33], thus making them suitable for wind turbine modelling including yawed configurations. The said models are often formulated as dynamic systems and their state derivatives are provided in the literature for implementation in solvers.

The reduced-order unsteady loads model considered in this work have their roots in the frequency domain studies from Theodorsen on unsteady attached flow aerodynamics [39] and the Beddoes model of dynamic stall [5]. Other dynamic stall methods, like the ONERA model [29], that uses parameters determined from experimental measurements on oscillating aerofoils, or the Snel model [37], that includes higher frequency dynamics of a self-excited nature,

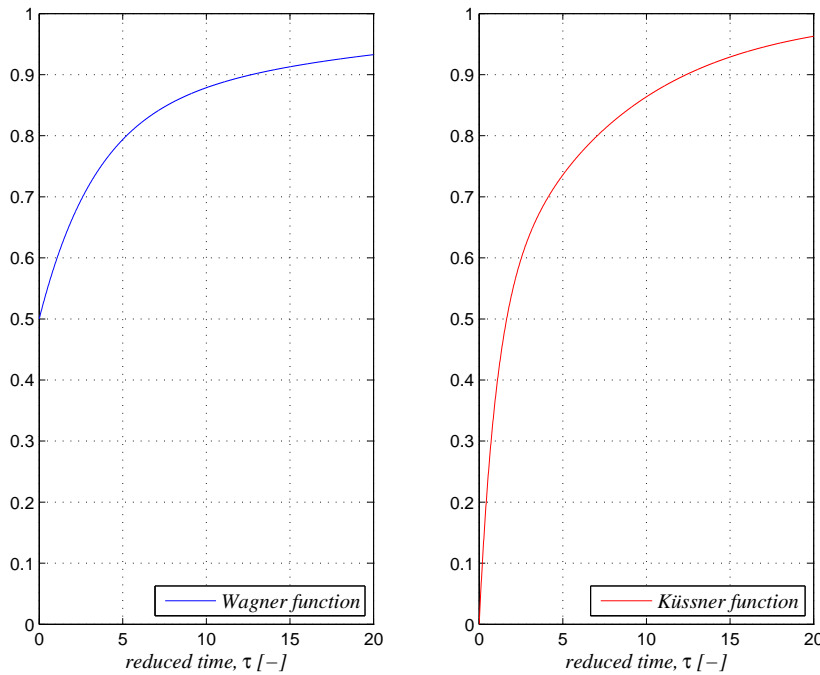


Figure 3: Wagner and Küssner function approximations. τ is the *reduced time*

have totally different origins and do not constitute prior art for the present work.

The Beddoes-Leishman semi-empiric methods for computation of unsteady aerodynamic loads reformulate the frequency domain Theodorsen theory into the time domain and propose a time integration of the Wagner function (figure 3) either by the Duhamel Integral or in terms of a state-space model [24, 25, 9]. The Duhamel integral helps coupling the model to time integrators and state-space representations are used for stability analysis and can be coupled to a comprehensive model by co-simulation or by adding the model equations to the holistic system.

2.2.1 Numerical Approximation to the Duhamel Integral

In the present, dynamic systems with arbitrary load histories are frequently solved, after several simplifying assumptions, by means of a recursive formulation of the Duhamel Integral. Based on the assumptions made, different final formulae are found and different accuracy figures are reached. The recursive nature of this method make it very useful for time integration of models with an arbitrary input and therefore they can be easily coupled to time integrators, given some minimal data exchange capabilities.

The indicial response of the aerodynamic coefficients to a step change in angle of attack was experimentally studied by Wagner in 1925 and obtained an exact formulation to the so called Wagner function (figure 3). A handy approximation to this curve was given by Jones in 1938 [16] in terms of a summation of exponential functions using the non-dimensional *reduced time*, $\tau = 2Vt/c$ which measures time as the number of semi-chords travelled at the airflow speed

$$\phi(\tau) = 1 - A_1 e^{-b_1 \tau} - A_2 e^{-b_2 \tau} \quad (2)$$

After neglecting short-term transient contributions, the following algorithm is obtained to find the time-varying value of the lift coefficient

$$C_l(\tau) = C_{l_\alpha} \alpha_E(\tau) \quad (3)$$

considering two deficiency functions for the equivalent angle of attack

$$\alpha_E(\tau) = \alpha(\tau) - X_1(\tau) - X_2(\tau) \quad (4a)$$

$$X_i(\tau) = X_i(\tau - \Delta\tau)e^{-b_i\Delta\tau} + I_i \quad (4b)$$

$$I_i = A_i e^{-b_i\tau} \int_0^\tau \frac{d\alpha}{d\tau}(\sigma) e^{b_i\sigma} d\sigma \quad (4c)$$

At this point, several algorithms exist based on the nature of the assumptions taken and yielding different accuracy figures. A first algorithm is based on a simple backward-difference approximation introduced for $d\alpha/d\tau \approx \Delta\alpha_\tau/\Delta\tau$ and time steps are taken small enough so that $b_i\Delta\tau$ are small. Then the deficiencies of angle of attack can be evaluated as

$$X_i(\tau) = X_i(\tau - \Delta\tau)e^{-b_i\Delta\tau} + A_i\Delta\alpha_\tau \quad (5)$$

A second more accurate algorithm is reached assuming $e^{b_i\sigma} \approx e^{b_i(\tau+\Delta\tau/2)}$, and based on the mid-point rule, so that

$$X_i(\tau) = X_i(\tau - \Delta\tau)e^{-b_i\Delta\tau} + A_i\Delta\alpha_\tau e^{-b_i\Delta\tau/2} \quad (6)$$

Similar procedures exist to integrate aerodynamic loads that derive from the Kssner function (figure 3) in which the change of angle of attack is progressive either due to its geometrical nature or due to compressibility effects.

2.2.2 State-space Models

In further works [25] Leishman and Nguyen obtain state-space representations based on the Wagner function approximation in terms of exponential summands. These formulations have many advantages over the original one; many dynamic models and control algorithms are represented in terms of ODE systems or state-space models and therefore they can be easily coupled together and they can be solved using well known iterative algorithms with user prescribed accuracy.

The time response to an arbitrary input history of a dynamic system can be formulated as the superposition of impulse responses $h(t)$ in the same fashion that it can be formulated as the superposition of indicial responses $\phi(t)$. Furthermore, it can be mathematically proven that the unit impulse response is the derivative of the unit step response.

$$h(t) = \frac{d}{dt}\phi(t) \quad (7)$$

The Laplace transform of the impulse response is normally called transfer function of the system $H(s)$, in s -domain. $H(s)$ is a mathematical representation of the quotient between the input $u(t)$ and output $y(t)$ of a linear time-invariant system

$$H(s) = \frac{Y(s)}{U(s)} \quad (8)$$

which can be represented with a state-space system (see appendix A) as in the following case with multiple input and multiple output variables

$$\left. \begin{aligned} \dot{\vec{x}}(t) &= \mathbf{A}\vec{x}(t) + \mathbf{B}\vec{u}(t) \\ \vec{y}(t) &= \mathbf{C}\vec{x}(t) + \mathbf{D}\vec{u}(t) \end{aligned} \right\} \quad (9)$$

the equivalent multi-dimensional transfer function being (appendix A)

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (10)$$

A state-space representation in diagonal form is obtained which contains exactly the same information as equation 3

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -f_s b_1 & 0 \\ 0 & -f_s b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \alpha(t) \\ C_l(t) &= C_{l\alpha} f_s \begin{bmatrix} A_1 b_1 & A_2 b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{C_{l\alpha}}{2} \alpha(t) \end{aligned} \quad (11)$$

where $f_s = \frac{2V}{c}$ is the flow rate in terms of number of semi-chords per second.

2.3 Coupled Simulation Methodologies

As simulation requirements and systems' complexity have grown during the past few decades, preexisting specialised simulation tools have integrated new multi-physics capabilities or have developed new environments to ease co-simulation schemes. Multipurpose commercial FE tools as ANSYS, SAMCEF, COMSOL or ABAQUS have enriched their packages with multiphysics libraries, solvers and applications.

Besides, researchers have struggled between tight coupled models seeking higher accuracy and lower CPU times, and loosely coupled models taking advantage of modularity and specialised solvers. Two scientific fields stand out in number of publications reporting multiphysics issues and strategies, namely Multibody Systems coupled to other physically coupled phenomena and continuum multiphysics simulations involving large mesh discretisations, such as CFD-CSD modelled fluid-structure interactions.

Published work in many different disciplines, however, still distinguish between two preferred approaches for mixed-domain problems: integrating secondary models into the main model in a tight coupling and solve the whole system at once or creating an ad-hoc interface and use the two or more models with their own solvers in a loose coupling co-simulation scheme [41, 1, 21, 36].

In a tight coupling scheme the whole system is solved in one process and the subsystems share unconverged displacements and forces. The process continues until equilibrium is reached by forces and displacements of all subsystems at once. In a loose coupling each subsystem is recalculated using subsystems converged outputs and reaching equilibrium by an iterative procedure of recalculations [11].

Tight coupling solutions are preferable when computation time and accuracy are required. However, tight coupling simulations are not always possible either because the tight interaction between modules is not known or because the main model simulation system is not open enough for the analyst to access and complement the system of equations. On the other hand the loosely coupling approach results in highly modular, is compatible with black box specialised solvers, provided minimum input-output capabilities.

As a trend, loose coupling co-simulation is increasingly chosen by analysts to tackle mixed-domain computation problems thanks to the increasing of computation speeds and to take advantage of specialised software packages. In this context, commercial software tools have added platforms for co-simulation included in their packages and an industrial-academic effort was done recently to standardise a co-simulation environment. The Functional Mock-up Interface (FMI) [11] developed under the ITEA 2 European Project is a recent effort to create a standard to ease co-simulation of models defined in several different

tools or platforms. The state of the art of co-simulation pointing in the direction of better communication among master and slave simulators, master algorithms and step management.

Academic teams and collaborative research programmes, however, are more up to the challenge of tight coupling schemes that involve more complex algorithms. A recent review by Keyes [19] reports the main present issues and future challenges of multiphysics simulations in a varied set of research fields like fluid-structure interaction, heat transfer with neutron transport in nuclear reactors and Multiscale methods, among others. State of the art of tight coupling is related to solver strategies including variations of the Newton method [19, 20].

3 Aim of Present Work

The aim of the present work is to provide a robust method to couple a dynamic system to an existing numerical model equipped with a time integration scheme. A particular objective of this paper is to apply this method to couple an in-house unsteady aerodynamic model with a Flexible Multibody model of rotor while maintaining convergence and CPU time standards as well as getting the most accurate results possible.

A closed-form expression for the exact solution of an ODE system coupled in a time integration scheme is presented and it is applied to solve the model of unsteady inflow velocities model, based on an ODE system, and to solve the 2D unsteady aerodynamic loads at aerofoil level, based on a state-space representation, both coupled to a Flexible Multibody rotor model.

The formula is also useful for any other phenomena that can be modelled by means of a dynamic system and it is to be solved coupled to a time integration scheme that provides for every time step the set of old forces and deformations and the set of predicted deformations for the present time step. Furthermore, the presented method is aimed to provide a robust and systematic way to couple an in-house model to an existing commercial code without compromising the stability of the holistic model and contributing to keep CPU time low.

4 Progress Beyond the State of the Art

As shown in section 2.1, many simulation techniques exist to help the analyst obtain time history of displacements, velocities and loads of complex numerical models. However, significant trouble may arise when these techniques are used incautiously and a number of linked methods must work properly to obtain reliable final results. For the present case, those links include the nature of both the multibody solver and the in-house unsteady aerodynamic model and the means to couple them.

The Duhamel integral approximation discussed in section 2.2.1 is a widely used approach to the problem but a simple recursive formulation is not always available or it requires a previous significant effort to formulate the assumptions that make the problem computationally affordable. On the other hand, the set of assumptions entails an approximation error that is more significant at bigger time step values and may lead to significant errors for long simulation load cases (table 1).

A harsh solution is always possible (see section 2.3) given enough computing force by coupling the two models in a co-simulation scheme. However, the converged partial results shared at the end of each iteration may not always be compatible with a solution of another part of the model. Nonetheless, it is indicated in the literature that attention should be paid to the mathematical aspects of performance and stability problems of co-simulation with different dynamic models [42]. Furthermore, coupling of simulators may result in an unstable integration, if an algebraic loop exists between the subsystems [21]. A

particular case of numerical instability may appear when at a certain operation point, the solution jumps between two configurations of the system at every Newton-Raphson iteration. The error at one side of the coupled system may be amplified by the dynamics of the other side and therefore leading to convergence trouble. This problem is magnified when conditional constructs are present. The unsteady loads model used in this work is an in-house Beddoes-Leishman-like model that aims to capture aerofoil performance. This is not limited to attached flow but also includes stalled conditions and, therefore, the set of flow separation, vortex shed over the aerofoil and other triggered phenomena are modelled using conditional constructs in the code.

In the case of an in-house model coupled to an existing time integrator as the one presented in this work, a numerically optimal solution is to append the in-house model equations to the existing ones taking advantage of the solver robustness in a fully coupled scheme but this is normally not the best option in commercial tools.

4.1 Proposed Solution to Couple Dynamic Systems

The method explained in this section is meant for dynamic systems that are solved in a time-discretised environment. Even if this is the most common representation it is interesting to show that the method can be also applicable to any model based on the principle of superposition of indicial responses by means of the Duhamel integral. By a previous exponential approximation of the indicial function the method can be formulated as a totally equivalent state-space representation without losing information.

Each scalar transfer function of the system, in s -domain, can be directly derived from the associated indicial response by using the Laplace transformation rules:

$$H_{i,j}(s) = s\mathcal{L}\{\phi_{i,j}(t)\} - \phi_{i,j}(0) \quad (12)$$

Since any state-space formulation has a transfer function associated,

$$\rightarrow \mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (13)$$

where \mathbf{A} is the dynamic matrix of the state-space, \mathbf{B} the control matrix, \mathbf{C} the observables matrix and \mathbf{D} the feed-forward matrix. At least one representation of the state-space can be easily found by simple manipulation of $\mathbf{H}(s)$ and term identification

$$\left. \begin{aligned} \dot{\vec{x}}(t) &= \mathbf{A}\vec{x}(t) + \mathbf{B}\vec{u}(t) \\ \vec{y}(t) &= \mathbf{C}\vec{x}(t) + \mathbf{D}\vec{u}(t) \end{aligned} \right\} \quad (14)$$

Finally, given \mathbf{A} is diagonalisable, a similarity transformation is applied to the original state-space representation so that the new representation has a diagonal \mathbf{A} matrix [30]. This means that the new states form a decoupled set and the dynamic equation can be treated as a set of simple ODEs.

$$\vec{z} = \mathbf{P}\vec{x} \quad (15)$$

$$\begin{aligned} \mathbf{A}_x &= \mathbf{P}^{-1}\mathbf{A}_z\mathbf{P}, & \mathbf{B}_x &= \mathbf{P}^{-1}\mathbf{B}_z \\ \mathbf{C}_x &= \mathbf{C}_z\mathbf{P}, & \mathbf{D}_x &= \mathbf{D}_z \end{aligned} \quad (16)$$

The state-space solution $\vec{y}(t)$ is of course invariant under this transformation and it is a direct contributor of the unsteady aerodynamic force. The result of

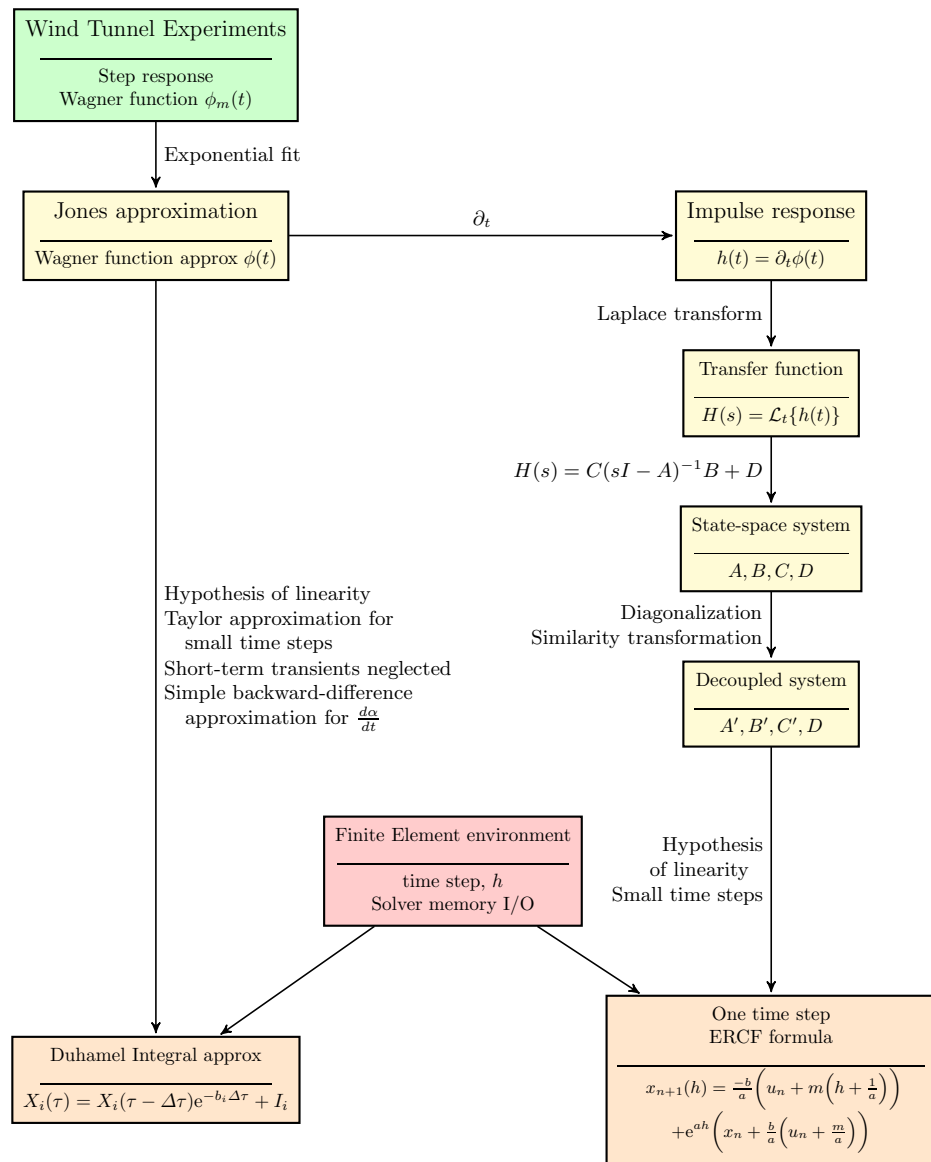


Figure 4: Mathematical procedure to transform an indicial response into the ERCF formula as an alternative to the Duhamel Integral approximation

the said formulation is a *de facto* linearised representation of the non-linear dynamic model that can be directly incorporated to a FE model and be processed using a linearised input function so that the whole system is solved simultaneously by the FE Newton-Raphson strategy. Since the non-linear dynamic system formulation is not restricted to any field, the presented method could be regarded, using the terminology by Keyes [19], as a solver method for multiphysics coupling that avoids nested iterations and takes advantage of the FE package solver.

4.1.1 Solution of an ODE for an Arbitrary Time History

The fact that the computation is coupled to a Flexible Multibody environment results in the inputs of the state-space model (chosen among system coordinates and their derivatives) being determined beforehand, i.e. coordinates (and their derivatives) are known for t^n and t^{n+1} .

Figure 5 shows known and unknown data in the time integration algorithm. Black solid line represents the actual trajectory of the system coordinates. It is known both for t^n and t^{n+1} and are inputs to the Exact Recursive Closed-Form expression (ERCF) formula for uncoupled linearised-input equations. Black

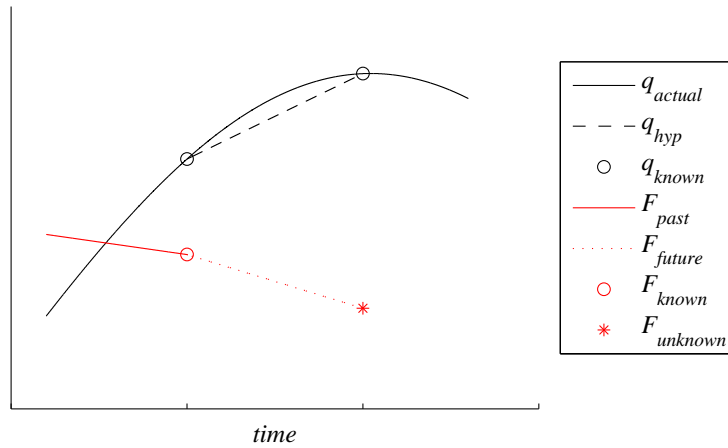


Figure 5: Scheme of known (circular marker) and unknown (cross marker) data in the time integration algorithm. Red cross represents the output forces computed at t^{n+1} using the ERCF formula

dashed line describes the hypothetical linear trajectory of the system coordinates between t^n and t^{n+1} . Red solid line describes the history of forces up to t^n , which is the output of the method. The red cross is the forces to be computed at t^{n+1} using the exact solution of the system when following the hypothetical trajectory described by the black dashed line.

Therefore, for a small time step value h , a simple evolution of coordinate vector in the time step range can be hypothesised, e.g. a linear time variation. Under this assumption an analytical recursive solution can be found for each decoupled state and expressed as function of state-space input values, state old values and time step. For each decoupled ODE,

$$\dot{x} - ax = bu(t) \tag{17}$$

trajectory of the system coordinates is assumed to be linear with time and it, therefore, can be built using known data from previous and actual time step,

$$u(t) = u^n + m(t - t^n) \tag{18}$$

with a constant rate of change

$$m = \frac{u_{n+1} - u_n}{t_{n+1} - t_n} \tag{19}$$

and the time dependency of a state is obtained from the solution of an ODE with a simple enough known source term

$$x(t) = e^{at} \left(\int_0^t e^{-as} b(u^n + m(s - t^n)) ds + C \right) \tag{20}$$

from which the ERCF formula is derived for the ODE when integrated between t^n and t^{n+1} assuming a linear trajectory of the ODE input variables

$$x^{n+1}(h) = \frac{-b}{a} \left(u^n + m \left(h + \frac{1}{a} \right) \right) + e^{ah} \left(x^n + \frac{b}{a} \left(u^n + \frac{m}{a} \right) \right) \tag{21}$$

Figure 6: Input source term $u(t)$ with increasing frequency

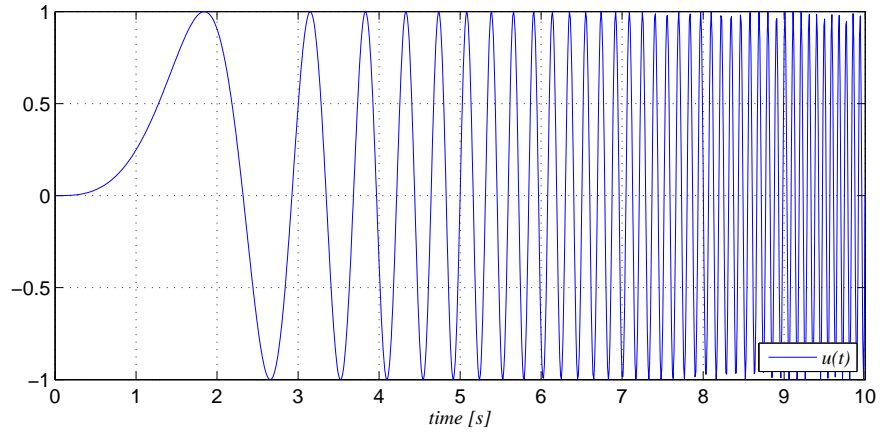
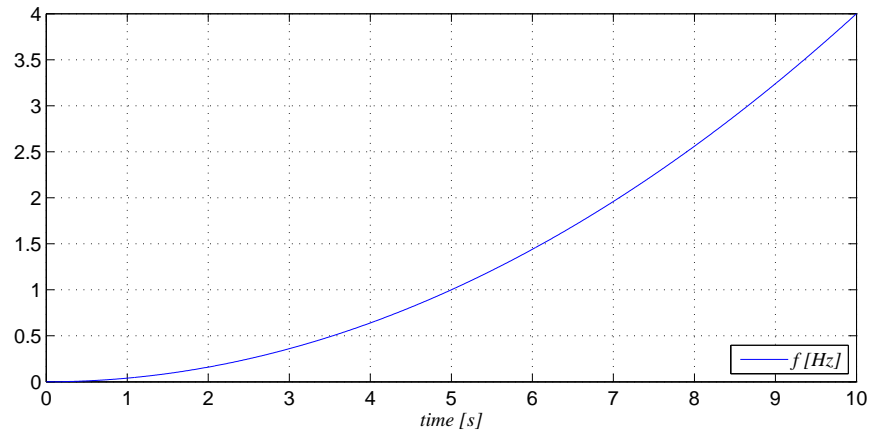


Figure 7: Input source term frequency schedule



4.2 Verification of the Method: a Simple Case

The method has been explored with the aim to verify it. It has been used to solve an example of state-space with a single state.

$$\left. \begin{aligned} \dot{x}(t) &= ax(t) + bu(t) \\ y(t) &= cx(t) + du(t) \end{aligned} \right\} \quad (22)$$

with the state-space constants

$$\begin{aligned} a &= -5, & b &= 2 \\ c &= 1, & d &= -8 \end{aligned} \quad (23)$$

and initial conditions

$$x(0) = 0 \quad (24)$$

to which an oscillating signal with increasing frequency is applied as input (see figures 6 and 7).

The system has been solved using the Matlab internal tool *lsim* and with the presented method, both with a time step of 0.01 s and plotted in figure 8.

4.3 Numerical Error

Error obtained at time step scale for the loads computation using the present method remains at the machine error level. Figure 9 shows the error dependency

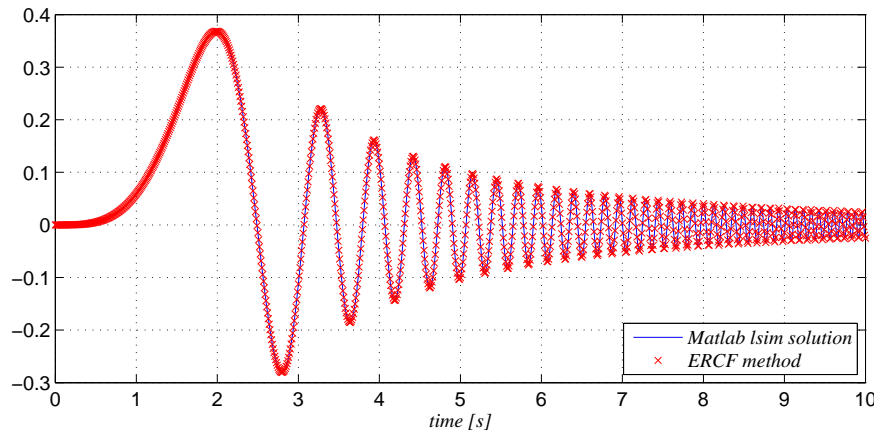


Figure 8: Output of the single state of a state-space. Blue line shows Matlab *lsim* command results. Red points show the ERCF results

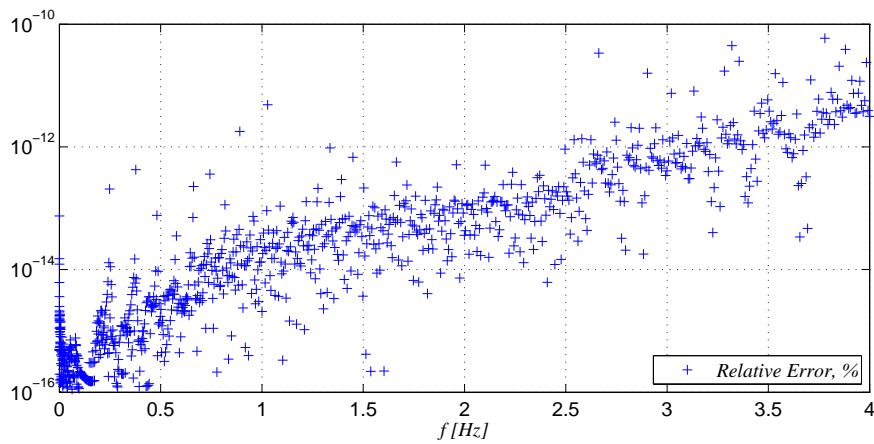


Figure 9: Relative error between Matlab *lsim* solution and the presented ERCF recursive formulation. Error values in % and represented with respect to input frequency

with frequency for the simple case explained above.

$$\epsilon = \frac{\|x_{lsim} - x_{ERCF}\|}{\|x_{lsim}\|} \quad (25)$$

5 Results

In rotor aerodynamics two important sources of unsteadiness exist, namely aerodynamic loads under arbitrary changes of apparent wind characteristics and incoming wind speed caused by arbitrary changes in the rotor loading. The first accounts for relatively fast, aerofoil scale, 2D phenomena including air circulation and dynamic stall; the second accounts for relatively slow air stream deceleration and deflection at rotor scale.

As said in section 3, one objective of the present work is to couple the said aerodynamic model with a Flexible Multibody tool. The resulting mixed-domain code will be used to capture aeroelastic phenomena and compute resulting loads. Many commercial aeroelastic codes combine multibody and FEM in one or another way [34, 17], including the open-source code MBDyn [28]. In the present work a full FEM flexible multibody code is chosen so that non-linear elasticity, kinematics and dynamics are well represented. The commercial code SAMCEF-MECANO was chosen for being a well validated code featuring high degree of openness to build custom models at several different levels. A comprehensive aeroelastic model of a wind turbine rotor was created comprising structural parts modelled as beams or more complex structures as Superelements [8]. Mechanical joints like bearings and drives are modelled as a series of lumped mechanism elements, such as hinges, springs and other one-dimensional

elements for specific functions as silent-blocks and local forces. The tool is chosen for its openness in the selection of elements, its connectivity and the possibility to add totally customised user elements or subsystems. Such a feature requires a totally merged communication with the solver and, at the same time, the access for the analyst to partial results during the iteration process including unconverged coordinates and rotations, storage vectors, computation time, iteration number and algorithm flags among others.

The SAMCEF-MECANO USER element has been used to couple an aerodynamic model with around 700 states for the aerodynamic loads model and 3 states for the dynamic inflow model with a Flexible Multibody model of the rotor with around 800 DOFs.

5.1 Unsteady Aerodynamic Loads

Originally formulated as an indicial problem and numerically solved as Duhamel integration of Wagner functions, the Leishman formulation for unsteady aerodynamic load is frequently expressed as a state-space model.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -f_s^2 b_1 b_2 & -f_s(b_1 + b_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \alpha(t) \\ C_l(t) &= C_{l\alpha} \begin{bmatrix} \frac{1}{2} f_s^2 b_1 b_2 & f_s(A_1 b_1 + A_2 b_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{C_{l\alpha}}{2} \alpha(t) \end{aligned} \quad (26)$$

where $f_s = \frac{2V}{c}$ is the flow rate in terms of number of semi-chords per second.

In this case a two-exponential summands function is used for the Jones approximation to the Wagner function

$$\phi_\alpha^C(t, M) = 1 - \sum_{i=1}^N A_i e^{-f_s b_i \beta^2 t} \quad (27)$$

A similarity transformation leads to a diagonal representation of the state-space model with two uncoupled states

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -f_s b_1 & 0 \\ 0 & -f_s b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \alpha(t) \\ C_l(t) &= C_{l\alpha} f_s \begin{bmatrix} A_1 b_1 & A_2 b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{C_{l\alpha}}{2} \alpha(t) \end{aligned} \quad (28)$$

Now, an exact solution can be found for each state assuming a linear time-variation of the source term $\alpha(t)$. Since this model is coupled to a FEM environment the source term of the loads model are known values of coordinates and their derivatives at t^n and t^{n+1} , as it is plot in figure 5. Therefore,

$$x_1^{n+1} = \frac{1}{f_s b_1} \left(\alpha^n + m_1 \left(h - \frac{1}{f_s b_1} \right) \right) + e^{-f_s b_1 h} \left(x_1^n - \frac{1}{f_s b_1} \left(\alpha^n - \frac{m_1}{f_s b_1} \right) \right) \quad (29)$$

$$x_2^{n+1} = \frac{1}{f_s b_2} \left(\alpha^n + m_2 \left(h - \frac{1}{f_s b_2} \right) \right) + e^{-f_s b_2 h} \left(x_2^n - \frac{1}{f_s b_2} \left(\alpha^n - \frac{m_2}{f_s b_2} \right) \right) \quad (30)$$

with source term variation rate being

$$m_1 = m_2 = \frac{\alpha^{n+1} - \alpha^n}{h} \quad (31)$$

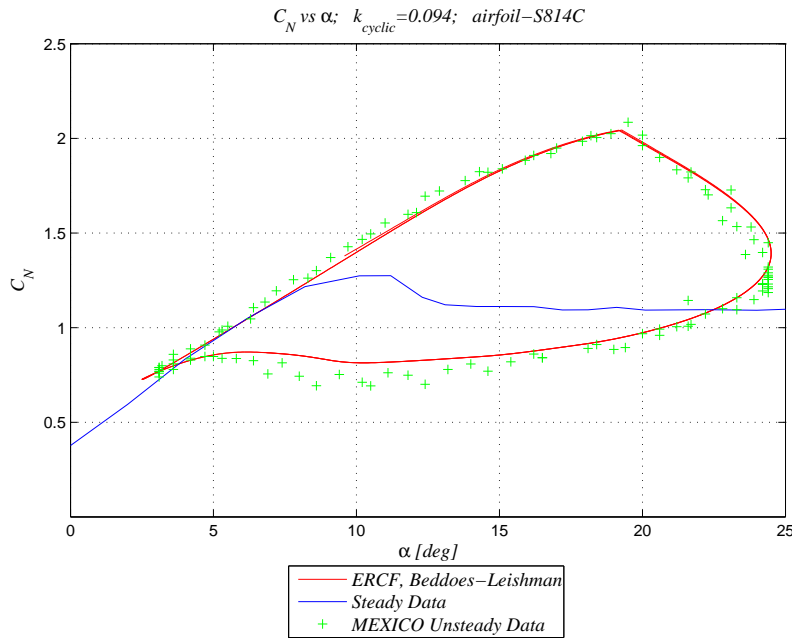


Figure 10: Beddoes-Leishman model results compared to MEXICO experiments. Steady C_N values are plot in blue; unsteady measurements are shown in green; ERCF results are shown in red

In a first phase of this work, the present formulation has been used to solve the Beddoes-Leishman unsteady dynamic model in stand-alone 2D aerofoil with the aim to validate it. In this case public data from the MEXICO experiments were used [32, 38]. Figure 10 shows C_N coefficient of a S814C aerofoil under sinusoidal oscillations of the angle of attack with a reduced frequency $k = 0.094$ in the deep stall region. Both unsteady measured and computed, as well as static results are presented.

5.2 Unsteady Inflow Model

Pitt-Peters is a very common algorithm for inflow wind speed field computation [33]. It computes the wind speed field on the rotor disc as a modification of the unperturbed wind speed field upstream. It captures wind direction to rotor axis misalignment as well as the unsteadiness of wind direction changes.

As expressed in [10] the Pitt-Peters inflow model is expressed as a three-DOFs coupled ODE system, one of them referring to the mean axial inflow and the other two referring to the two linear contributions along the the mutually orthogonal coordinates in the non-rotating disk.

$$\mathbf{M} \begin{bmatrix} \dot{v}_0 \\ \dot{v}_s \\ \dot{v}_c \end{bmatrix} + \mathbf{L}^{-1} \begin{bmatrix} v_0 \\ v_s \\ v_c \end{bmatrix} = \begin{bmatrix} C_T \\ C_l \\ C_m \end{bmatrix} \quad (32)$$

where the so called gain matrix \mathbf{L} is

$$\mathbf{L} = \begin{bmatrix} \frac{1}{2V_T} & 0 & \frac{15\pi}{64V_m} \tan\left(\frac{\chi}{2}\right) \\ 0 & \frac{-4}{V_m(1+\cos\chi)} & 0 \\ \frac{15\pi}{64V_T} \tan\left(\frac{\chi}{2}\right) & 0 & \frac{-4 \cos \chi}{V_m(1+\cos \chi)} \end{bmatrix} \quad (33)$$

and the apparent mass matrix \mathbf{M} for twisted-blade rotors is

$$\mathbf{M} = \begin{bmatrix} \frac{128}{75\pi} & 0 & 0 \\ 0 & \frac{-16}{45\pi} & 0 \\ 0 & 0 & \frac{-16}{45\pi} \end{bmatrix} \quad (34)$$

When implemented in the finite element environment, this model is solved as a system of Ordinary Differential Equations

$$\dot{\vec{v}} + \mathbf{A}\vec{v} = \vec{b} \quad (35)$$

the ODE system matrix being

$$\mathbf{A} = \mathbf{M}^{-1}\mathbf{L}^{-1} \quad (36)$$

and assuming a linear time-variation of the source term \vec{b} between t^n and t^{n+1}

$$\begin{aligned} \vec{b}^n &= \mathbf{M}^{-1}\vec{C}^n \\ \vec{b}^{n+1} &= \mathbf{M}^{-1}\vec{C}^{n+1} \end{aligned} \quad (37)$$

Then, the values of the dynamic inflow factors can be written as an analytical, closed form for the next time step

$$\begin{aligned} v_0^{n+1} &= F_0 + G_0h + H_0 e^{-a_{1,1}h} + v_0^n e^{-a_{1,1}h} \\ v_s^{n+1} &= F_s + G_s h + H_s e^{-a_{2,2}h} + v_s^n e^{-a_{2,2}h} \\ v_c^{n+1} &= F_c + G_c h + H_c e^{-a_{3,3}h} + v_c^n e^{-a_{3,3}h} \end{aligned} \quad (38)$$

with the constants

$$\begin{aligned} F_0 &= \frac{b_1^n - G_0}{a_{1,1}}; & G_0 &= \frac{b_1^{n+1} - b_1^n}{a_{1,1}h}; & H_0 &= -F_0 \\ F_s &= \frac{b_2^n - G_s}{a_{2,2}}; & G_s &= \frac{b_2^{n+1} - b_2^n}{a_{2,2}h}; & H_s &= -F_s \\ F_c &= \frac{b_3^n - G_c}{a_{3,3}}; & G_c &= \frac{b_3^{n+1} - b_3^n}{a_{3,3}h}; & H_c &= -F_c \end{aligned} \quad (39)$$

6 Discussion and Conclusions

6.1 Discussion of the Method

6.1.1 Numerical Errors

As mentioned in paragraph 2.2.1, several different algorithms exist that implement numerical solutions of the Duhamel Integral applied to the computation of unsteady aerodynamic loads. Depending on the nature of assumptions taken to evaluate the integral I_i in equation 4c, different algorithms with different accuracy can be derived. A first algorithm is based on a simple backward-difference approximation introduced for $d\alpha/d\tau \approx \Delta\alpha_\tau/\Delta\tau$. Thus,

$$I_i = A_i \frac{\Delta\alpha_\tau}{\Delta\tau} \left(\frac{1 - e^{-b_i \Delta\tau}}{b_i} \right) \quad (40)$$

	Wagner function		Kssner function	
	Blade root	Blade tip	Blade root	Blade tip
ϵ_1	-1.9%	-29%	-6.3%	-110%
ϵ_2	0.005%	0.96%	0.05%	9.9%

Table 1: Duhamel Integral relative errors at two stages of a 110 m diameter wind turbine blade

and time steps are taken small enough so that $b_i \Delta\tau$ are small. Then, it is assumed that

$$\frac{1 - e^{-b_i \Delta\tau}}{b_i} \approx \Delta\tau \tag{41}$$

yielding the following recursive formula ready to implement in numerical codes

$$X_i(\tau) = X_i(\tau - \Delta\tau)e^{-b_i \Delta\tau} + A_i \Delta\alpha_\tau \tag{42}$$

The relative error due to the assumptions made depends on unsteady aerodynamic parameters and time step. Practical error values are found in table 1 for unsteady phenomena based on Wagner function and Kssner function when zero instantaneous response is desired.

$$\epsilon_1 = 2 - \frac{b_1 \Delta\tau}{1 - e^{-b_1 \Delta\tau}} - \frac{b_2 \Delta\tau}{1 - e^{-b_2 \Delta\tau}} \tag{43}$$

A second more accurate algorithm is reached when assuming $e^{b_i \sigma} \approx e^{b_i(\tau + \Delta\tau/2)}$, and based on the mid-point rule. After some mathematical manipulation, an alternative recursive formula is obtained

$$X_i(\tau) = X_i(\tau - \Delta\tau)e^{-b_i \Delta\tau} + A_i \Delta\alpha_\tau e^{-b_i \Delta\tau/2} \tag{44}$$

with a lower relative error (see table 1)

$$\epsilon_2 = 2 - \frac{b_1 \Delta\tau e^{-b_1 \Delta\tau/2}}{1 - e^{-b_1 \Delta\tau}} - \frac{b_2 \Delta\tau e^{-b_2 \Delta\tau/2}}{1 - e^{-b_2 \Delta\tau}} \tag{45}$$

Taking on account that this work is thought to be part of a simulation tool using a variable time step with an upper limit at $h_{MAX} = 0.01$ s, the aerodynamic loads computation error depends on the geometry and the rotor working conditions. For a 110 m diameter wind turbine rotating at around 13 rpm, the error can easily surpass $\epsilon = 1\%$ in the best case in one single time step. Computation of 600 s time simulations, as in wind energy industry standards, suggests the need to lower these error figures.

6.1.2 Computation Time of the Comprehensive Model

Guidelines for certification of wind turbines require the simulation of around one hundred 600 s time stories with turbulent wind plus around 400 load cases of 100 s and 200 s simulation times with singular events. Using a 32 bit Windows system on a Pentium i7 computer, a typical full certification procedure can take near 300 h of CPU time. The same machine is used to assess the computation times involved in the solution of the ODE system with both recursive and iterative methods. A version of the explicit Adams PECE solver [35], as distributed under the GNU LGPL license by Burkardt [7] and featuring user prescribed accuracy, was implemented and used to solve ODE systems of several sizes. Series of thousands of linear ODE systems of several dimensions and with random constants are programmed in Fortran 90 and run in this machine to evaluate average CPU times. A similar procedure is followed for the ERCF method. Both the ERCF method presented here and the numerical approximation to

Table 2: Computation times of a baseline S4WT model and additional CPU time of unsteady aerodynamic 8-states ODEs

Load case times, s	CPU time, min		Additional CPU time	
	Baseline S4WT ($h = 0.01$ s)	Adams PECE iterative solver	ERCF recursive formula	
100 s	16.5	+3.0 min	+0.9 s	
200 s	30.3	+6.0 min	+1.8 s	
600 s	82.5	+17.9 min	+5.5 s	

Table 3: CPU times of Adams PECE solver and ERCF method solving ODE systems of several sizes

ODES size Num of states	CPU time, μ s		
	Adams PECE solver	ERCF formula	Adams PECE/ERCF ratio
1	8.7	0.042	208
2	12.5	0.086	145
3	18.2	0.105	173
8	52.2	0.289	181
15	122.2	0.546	224

the Duhamel integration are recursive methods and contain similar number of operations and therefore yield similar CPU times .

With an absolute local error tolerance set to 10^{-5} , the computation time attributed to the 8-states ODE system solution for one iteration grows by a factor 200 with respect to the recursive methods (table 3), increasing the sum-up of aerodynamics computation times from the 5s of the recursive methods to 17min, which is a very significant contribution to a 82min of the baseline SAMCEF-MECANO case. Furthermore, the computation times using iterative solvers grows quadratically with the number of states, whereas it does linearly with the recursive methods.

In a comprehensive model with 8 states for each unsteady loads system computed at every blade station, with 30 stations per blade, and a 3 states inflow model for the rotor, the total computation time due to the solution of unsteady aerodynamic ODEs in a 600s time history using a $h = 0.01$ s time step which needs an average of 3.5 iterations per step, ranges from 16 min to 18 min depending on the amount of independent subsystems present in the case of very sparse systems.

Solving the same phenomena using a recursive closed-form representation like the presented method, reduces CPU times to approximately 5.5s for each 600s load case (table 4). For a full certification procedure with high accuracy, the recursive closed-form methods save up to 46 h of computation time. For the Duhamel integral, frequently used in the literature for such models, the use of mathematical operations and memory is very similar, therefore it is reckoned to yield computation savings of the same order.

6.2 Conclusions

A new method for coupling an unsteady aerodynamic model based on dynamic systems to a finite element based flexible multibody model of wind turbine has been presented.

Table 4: CPU times of Adams PECE solver and ERCF method solving the set of ODE systems present in the model for a 600s load case. The percentages of the total computation time are included in parentheses

ODES size Num of states	CPU time	
	Adams PECE solver	ERCF formula
8	15.6 min	5.5 s
	(16%)	(0.11%)
15	36.4 min	10.3 s
	(31%)	(0.21%)

The method is formulated as an exact closed solution of an ordinary differential equation with a known linear source term that is compatible with the unknowns and outputs of a one-step time integrator scheme.

The method is therefore valid to model any other phenomenon that can be modelled by means of an Ordinary Differential Equation or by an ODE system with a diagonalisable dynamic matrix so that it can be decoupled beforehand. Full efficiency is shown when the dynamic system is diagonalised symbolically beforehand.

The method is applicable to the integration of any stand-alone dynamic system with an arbitrary source term function, given a decoupled representation of the ODE system.

Results show very high accuracy, obtaining error figures in the limit of machine precision when comparing a stand-alone test case solved with the *lsim* tool from Matlab.

The method is used to solve an aeroelastic problem coupling an in-house 700 states unsteady aerodynamic model including unsteady inflow and unsteady loads with a 800 DOFs FEM based flexible multibody model of a wind turbine. The model showed excellent robustness and CPU times.

When compared to nested iterative algorithms to reach similar accuracy the exact closed-form method shows to be computationally much faster. CPU time savings are comparable to those reached with the numerical approximation of the Duhamel Integral found in the literature.

When compared to the numerical approximation to the Duhamel Integral, with errors around 5% and 1% depending on the assumptions made and the time step, the presented method shows the advantage of yielding machine precision accuracy.

Acknowledgement

This paper is based on a previous work done under the supervision of Marilena Pavel and Michel van Tooren at the Aerospace Engineering Faculty of the Technical University of Delft.

References

- [1] NR Aluru and Jacob White. A multilevel newton method for mixed-energy domain simulation of mems. *Microelectromechanical Systems, Journal of*, 8(3):299–308, 1999.
- [2] George B Arfken, Hans J Weber, and Frank E Harris. *Mathematical methods for physicists*. Academic press, 2005.
- [3] European Wind Energy Association. *Wind energy—the facts: a guide to the technology, economics and future of wind power*. Earthscan, 2009.
- [4] Olivier A Bauchau. Dymore users manual. *Georgia Inst. of Technology, Atlanta*, 2007.
- [5] TS Beddoes. Practical computation of unsteady lift. *Vertica*, 8(1):55–71, 1984.
- [6] EA Bossanyi. Bladed theory manual version 4.0. *Bristol, UK: Garrad Hassan and Partners Ltd*, 2010.
- [7] John Burkardt. Source codes in fortran90, August 2009.
- [8] Alberto Cardona and Michel Geradin. Modelling of superelements in mechanism analysis. *International Journal for Numerical Methods in Engineering*, 32(8):1565–1593, 1991.

- [9] Chawin Chantharasenawong. *Nonlinear aeroelastic behaviour of aerofoils under dynamic stall*. PhD thesis, Imperial College London (University of London), 2007.
- [10] Robert T.N Chen. A survey of nonuniform inflow models for rotorcraft flight dynamics and control applications. 1989.
- [11] MODELISAR Consortium et al. Functional mock-up interface for co-simulation. *Accessed March*, 1:2013, 2010.
- [12] Michel Géradin and Alberto Cardona. *Flexible multibody dynamics: a finite element approach*. John Wiley, 2001.
- [13] Global Wind Energy Council GWEC. Global wind statistics 2011, 2012.
- [14] Andreas Heege, Jaume Betran, and Yvan Radovicic. Fatigue load computation of wind turbine gearboxes by coupled finite element, multi-body system and aerodynamic analysis. *Wind Energy*, 10(5):395–413, 2007.
- [15] Wayne Johnson. A comprehensive analytical model of rotorcraft aerodynamics and dynamics. part 1. analysis development. Technical report, DTIC Document, 1980.
- [16] Robert T Jones. *The unsteady lift of a wing of finite aspect ratio*, volume 681. NACA, 1940.
- [17] J Jonkman and W Musial. Offshore code comparison collaboration (oc3) for iea task 23 offshore wind technology and deployment. *Contract*, 303:275–3000, 2010.
- [18] Jason M Jonkman and Marshall L Buhl Jr. Fast users guide. *Golden, CO: National Renewable Energy Laboratory*, 2005.
- [19] David E Keyes, Lois C McInnes, Carol Woodward, William Gropp, Eric Myra, Michael Pernice, John Bell, Jed Brown, Alain Clo, Jeffrey Connors, et al. Multiphysics simulations challenges and opportunities. *International Journal of High Performance Computing Applications*, 27(1):4–83, 2013.
- [20] Dana A Knoll and David E Keyes. Jacobian-free newton–krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.
- [21] R Kübler and W Schiehlen. Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamical Systems*, 6(2):93–113, 2000.
- [22] Torben J Larsen and Anders Melchior Hansen. *How 2 HAWC2, the user’s manual*. Risø National Laboratory, 2007.
- [23] J Gordon Leishman. Challenges in modelling the unsteady aerodynamics of wind turbines. *Wind energy*, 5(2-3):85–132, 2002.
- [24] JG Leishman and TS Beddoes. A semi-empirical model for dynamic stall. *Journal of the American Helicopter Society*, 34(3):3–17, 1989.
- [25] JG Leishman and KQ Nguyen. State-space representation of unsteady airfoil behavior. *AIAA journal*, 28(5):836–844, 1990.
- [26] Germanischer Lloyd. Rules and guidelines. *Wind Energy. Hamburg*, 2005.
- [27] Germanischer Lloyd. Rules and guidelines. *Wind Energy. Hamburg*, 2012.
- [28] Pierangelo Masarati. Mbdyn theory and developers manual version 1. x-devel, 2010.

- [29] Kenneth W McAlister, O Lambert, and D Petot. Application of the onera model of dynamic stall. Technical report, DTIC Document, 1984.
- [30] Katsuhiko Ogata. *Discrete-time control systems*, volume 1. Prentice-Hall Englewood Cliffs, NJ, 1987.
- [31] S Øye. Flex4. simulation of wind turbine dynamics. 1996.
- [32] Ricardo Pereira, Gerard Schepers, and Marilena D Pavel. Validation of the beddoes-leishman dynamic stall model for horizontal axis wind turbines using mexico data. *Wind Energy*, 2012.
- [33] David A Peters and Ninh HaQuang. Dynamic inflow for practical applications. 1988.
- [34] Wojciech Popko, Fabian Vorpahl, Adam Zuga, Martin Kohlmeier, Jason Jonkman, Amy Robertson, Torben J Larsen, Anders Yde, Kristian Sætertrø, Knut M Okstad, et al. Offshore code comparison collaboration continuation (oc4), phase i—results of coupled simulations of an offshore wind turbine with jacket support structure. In *22nd International Society of Offshore and Polar Engineers Conference. Rhodes, Greece, 2012*.
- [35] Lawrence F Shampine and Marilyn Kay Gordon. *Computer solution of ordinary differential equations: the initial value problem*, volume 1. Freeman San Francisco, 1975.
- [36] Rajarishi Sinha, Christiaan JJ Paredis, Vei-Chung Liang, and Pradeep K Khosla. Modeling and simulation methods for design of engineering systems. *J. Comput. Inf. Sci. Eng.*, 1(1):84–91, 2001.
- [37] H Snel. Heuristic modelling of dynamic stall characteristics. In *EWEC-Conference*, pages 429–433. Bookshop for Scientific Publications, 1997.
- [38] H Snel, JG Schepers, and B Montgomerie. The mexico project (model experiments in controlled conditions): The database and first results of data processing and interpretation. In *Journal of Physics: Conference Series*, volume 75, page 012 014. IOP Publishing, 2007.
- [39] Theodore Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. 1935.
- [40] J Thirstrup Petersen. The aeroelastic code hawc-model and comparisons. 1996.
- [41] Ondřej Vaculín, Wolf R Krüger, and Michael Valášek. Overview of coupling of multibody and control engineering tools. *Vehicle System Dynamics*, 41(5):415–429, 2004.
- [42] A Veitl, T Gordon, A Van De Sand, M Howell, M Valasek, O Vaculin, and P Steinbauer. Methodologies for coupling simulation models and codes in mechatronic system analysis and design. In *Vehicle System Dynamics*, 33: 231-243, 1999.
- [43] Tamer M Wasfy and Ahmed K Noor. Computational strategies for flexible multibody systems. *Applied Mechanics Reviews*, 56(6):553–614, 2003.

Appendices

A Transfer Function of a State-Space

A state-space formulation has a transfer function associated:

$$\left. \begin{aligned} \dot{\vec{x}}(t) &= \mathbf{A}\vec{x}(t) + \mathbf{B}\vec{u}(t) \\ \vec{y}(t) &= \mathbf{C}\vec{x}(t) + \mathbf{D}\vec{u}(t) \end{aligned} \right\} \quad (46)$$

where \mathbf{A} is the dynamic matrix, \mathbf{B} the control matrix, \mathbf{C} the observables matrix and \mathbf{D} the feed-forward matrix.

Operating in Laplace domain:

$$\left. \begin{aligned} s\vec{X}(s) &= \mathbf{A}\vec{X}(s) + \mathbf{B}\vec{U}(s) \\ \vec{Y}(s) &= \mathbf{C}\vec{X}(s) + \mathbf{D}\vec{U}(s) \end{aligned} \right\} \quad (47)$$

$$\rightarrow \vec{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\vec{U}(s) \quad (48)$$

$$\rightarrow \vec{Y}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\vec{U}(s) + \mathbf{D}\vec{U}(s) \quad (49)$$

$$\rightarrow \mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (50)$$

$\mathbf{H}(s)$ is the matrix of transfer functions among every input and every output.

The denominator of the transfer function is the characteristic polynomial $\lambda(s) = \det(s\mathbf{I} - \mathbf{A})$, therefore, matrix \mathbf{A} of a factorized denominator can be written as a diagonalized matrix.

A.1 Diagonal Form of a Leishman Loads Model

The state equations describing the behaviour of the unsteady aerofoil in attached flow conditions can be obtained by directly application of Laplace transform of the indicial response.

In this paragraph a simple case of unsteady lift in arbitrary angle of attack changes is treated without initial conditions (Kssner effect) in diagonal form.

From a transfer function the lift response to an input $\alpha(t)$ can be directly written in state-space form.

$$H(s) = \frac{A_1 b_1 f_s (s + b_2 f_s) + A_2 b_2 f_s (s + b_1 f_s)}{(s + b_1 f_s)(s + b_2 f_s)} \quad (51)$$

According to the state-space block diagram

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \quad (52)$$

the following states vector is selected:

$$\vec{x} = \begin{bmatrix} 1/(s + f_s b_1) \\ 1/(s + f_s b_2) \end{bmatrix} \quad (53)$$

For each output variable, matrix \mathbf{C} can be derived watching the transfer function written as a linear combination of states and the input variables. Thus, matrix \mathbf{C} is made of as much rows as outputs are there in the system and each row is the constants vector of the corresponding output variable.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -f_s b_1 & 0 \\ 0 & -f_s b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \alpha(t) \\ C_l(t) &= C_{l\alpha} f_s [A_1 b_1 \quad A_2 b_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (54)$$

A.2 Canonical Form of a Leishman Loads Model

In this paragraph a simple case of unsteady lift in arbitrary angle of attack changes is treated without initial conditions in canonical form.

The transfer function in the paragraph above can also be written as follows (using $A_1 + A_2 = 1$ found above):

$$H(s) = \frac{f_s A_1 b_1 (s + f_s b_2) + f_s A_2 b_2 (s + f_s b_1)}{(s + f_s b_1)(s + f_s b_2)} \quad (55)$$

$$H(s) = f_s^2 b_1 b_2 \frac{1}{s^2 + f_s (b_1 + b_2) s + f_s^2 b_1 b_2} + f_s (A_1 b_1 + A_2 b_2) \frac{1}{s^2 + f_s (b_1 + b_2) s + f_s^2 b_1 b_2} \quad (56)$$

From this second representation of the transfer function an alternative representation in the state-space can be formulated: the controllable canonical form.

$$\vec{x} = \begin{bmatrix} x_1 \\ \dot{x}_1 \end{bmatrix} \quad (57)$$

Using the derivative in the s domain (Laplace), which is simply multiplying by s , the new states vector can be derived by inspection:

$$\vec{x}(s) = \begin{bmatrix} x_1(s) \\ s x_1(s) \end{bmatrix} = \begin{bmatrix} 1/(s^2 + f_s (b_1 + b_2) s + f_s^2 b_1 b_2) \\ s/(s^2 + f_s (b_1 + b_2) s + f_s^2 b_1 b_2) \end{bmatrix} \quad (58)$$

Therefore, matrix \mathbf{A} (i.e. the transfer function) contains the equation that relates both states $x_2 = x_1$ in row 1 and the hidden second order differential equation of x_1 in second row. This second order equation on x_1 contains all the information of the state-space, therefore it will take the form of the transfer function. Consistently, input variables do not contribute to the first equation.

$$x_2 = \dot{x}_1 \rightarrow x_2(s) = s x_1(s) \quad (59a)$$

$$\dot{x}_2 = \ddot{x}_1 \rightarrow s x_2(s) = s^2 x_1(s) \quad (59b)$$

$$s^2 x_1 = f_s (A_1 b_1 + A_2 b_2) s x_1 + f_s^2 b_1 b_2 x_1 \quad (59c)$$

therefore, the state-space system is represented as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -f_s^2 b_1 b_2 & -f_s (b_1 + b_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \alpha(t) \quad (60)$$

$$C_l(t) = C_{l\alpha} \begin{bmatrix} f_s^2 b_1 b_2 & f_s (A_1 b_1 + A_2 b_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

B One Time-step Formula of a Decoupled State-space Model

From [2] the solution of an ODE with a known excitation function

$$\dot{x} + p(t)x = q(t) \quad (61)$$

can be evaluated as

$$x(t) = e^{-\int_0^t p(s) ds} \left(\int_0^t q(s) e^{\int_0^s p(v) dv} ds + C \right) \quad (62)$$

Given a states equation

$$\dot{\vec{x}} = \mathbf{A}\vec{x} + \mathbf{B}\vec{u} \quad (63)$$

and its uncoupled representation $\mathbf{A}' = a'_{ii}$, $\mathbf{B}' = b'_{ik}$, the integral above can be solved for each state i of the new representation using some intentionally chosen constant b and input function $u(t)$.

$$p(t) = -a'_{ii} = -a \quad (64)$$

$$q(t) = \sum_k b'_{ik} u_k(t) = b u(t) \quad (65)$$

A linear trajectory of the source term $u(t)$ between t_n and t_{n+1} is chosen for simplicity, but higher degree curves can be used instead.

$$u(t) = u_n + m(t - t_n) \quad (66)$$

with

$$m = \frac{u_{n+1} - u_n}{t_{n+1} - t_n} \quad (67)$$

The said source term is used in the states equation and symbolic integration is done for $x(t)$

$$\dot{x} - ax = b u(t) \quad (68)$$

$$x(t) = e^{at} \left(\int_0^t e^{-as} b(u_n + m(s - t_n)) ds + K \right) \quad (69)$$

$$x(t) = \frac{-b}{a} \left(u_n + m \left(t - t_n + \frac{1}{a} \right) \right) + K e^{at} \quad (70)$$

the value of the states at the beginning of the time-step are known

$$x(t_n) = x_n = \frac{-b}{a} \left(u_n + \frac{m}{a} \right) + K e^{at_n} \quad (71)$$

$$\implies K = e^{-at_n} \left(x_n + \frac{b}{a} \left(u_n + \frac{m}{a} \right) \right) \quad (72)$$

finally, the value of the states are found both expressed with respect with time t and with time-step h .

$$x(t) = \frac{-b}{a} \left(u_n + m \left(t - t_n + \frac{1}{a} \right) \right) + e^{a(t-t_n)} \left(x_n + \frac{b}{a} \left(u_n + \frac{m}{a} \right) \right) \quad (73)$$

$$x(h) = \frac{-b}{a} \left(u_n + m \left(h + \frac{1}{a} \right) \right) + e^{ah} \left(x_n + \frac{b}{a} \left(u_n + \frac{m}{a} \right) \right) \quad (74)$$