

Pilot-in-the-loop simulation of flexible aircraft in MATLAB/Simulink: implementation and coding peculiarities

Guillermo Lopez,^{*} Vilius Portapas,[†] Gaetan Dussart[‡] and Mudassir Lone[§]

Cranfield University, Cranfield, MK43 0AL, United Kingdom

Etienne Coetzee[¶]

Airbus Operations, Bristol, BS99 7AR, United Kingdom

Integration of flight dynamic models, developed in the MATLAB[®]/Simulink[®] environment, with an engineering flight simulation platform allows rapid pilot-in-the-loop evaluation of new aircraft concepts at early stages of design. This paper aims to provide an overview of the integration activities needed to develop an engineering flight simulator capable of providing means to assess future concept aircraft, such as high aspect ratio wing configurations, where aeroelastic effects have a significant impact on rigid body flight dynamics. Details of the approach used to integrate an aeroelastic simulation framework with an engineering flight simulator are presented. The challenges of obtaining a real-time simulation capability and coding peculiarities of this approach are discussed. The paper expands on the discussion of integration and coding, and provides an example that demonstrates capabilities of such a framework for handling qualities assessment of a high aspect ratio wing aircraft.

I. Introduction

Numerous aircraft concepts with novel technologies¹⁻³ have been developed to address efficiency and environmental targets.^{4,5} Most of them have demonstrated the need for multidisciplinary integration of technologies and the use of novel engineering methods, which provide a rapid assessment capability of new configurations to evaluate them in early stages of design. Traditionally, a pilot-in-the-loop assessment of flight dynamics is carried out after fixing certain aspects of the design. However, new aircraft concepts such as blended wing bodies,⁶ which introduce flight dynamics that are still poorly understood, or high aspect ratio wing configurations,^{7,8} which introduce flight dynamics coupling with structural dynamics due to wing flexibility, require an early piloted assessment to find out potential issues with aircraft handling qualities. Hence, flight simulators are used for the pilot-in-the-loop evaluation of new aircraft configurations. Flight simulators are usually linked with pilot training. However, there are different types of flight simulators used for engineering and/or research purposes. They are known as engineering flight simulators and play a critical role in exploring aerodynamic and structural behaviour of new aircraft concepts as well as assessing new algorithms of flight control systems and their impact to the aircraft controllability and stability.⁹ Such a simulator has been recently developed at Cranfield University and the process of developing and integrating different tools required for such a flight simulator is presented in this paper.

The paper is structured in a way to provide a discussion on each component of the simulator. Section II provides a short discussion on the Cranfield Accelerated Aeroplane Loads Model (CA²LM) as its development

^{*}Research Assistant, Centre for Aeronautics, School of Aerospace, Transport & Manufacturing, g.lopez@cranfield.ac.uk.

[†]Research Student, Centre for Aeronautics, School of Aerospace, Transport & Manufacturing, v.portapas@cranfield.ac.uk.

[‡]Research Student, Centre for Aeronautics, School of Aerospace, Transport & Manufacturing, g.x.dussart@cranfield.ac.uk.

[§]Lecturer, Centre for Aeronautics, School of Aerospace, Transport & Manufacturing, m.m.lone@cranfield.ac.uk.

[¶]Future Projects Engineer, Airbus Operations, etienne.coetzee@airbus.com.

peculiarities and applications have been already discussed.^{10–14} Section III provides a discussion on hardware and software components of the Engineering Flight Simulator 500 (EFS500). The process of integrating the CA²LM framework with the EFS500 flight simulator is presented in Section IV. An application example of the newly developed flight testing framework is given in Section V. Finally, key findings of the integration process are concluded and further improvements to the simulator are suggested in Section VI.

II. Review of the CA²LM Framework

The CA²LM framework constitutes one of two main parts of the simulation framework by providing a MATLAB[®]/Simulink[®] based environment for the modelling and simulation of an aeroelastic aeroplane. Based on low fidelity aerodynamic and structural models and integrated with a flight simulator hardware it enables real-time pilot-in-the-loop simulations for handling qualities testing of highly flexible aeroplanes. This section provides a brief overview of aerodynamic and structural models and equations of motion for flexible aircraft modelling and reviews further developments of the model.

II.A. Aerodynamic model

Aerodynamic models of the wing and horizontal/vertical tailplane are based on the Modified Strip Theory (MST) steady aerodynamics coupled with the Leishman-Beddoes unsteady aerodynamic model.¹¹ To model the unsteady build-up of the lift due to changes in the angle of attack and airspeed, a state-space representation of the unsteady aerodynamics of the aerofoil has been implemented following the work done by Leishman and Nguyen.¹⁵ The CA²LM framework uses two-pole approximation of the Wagner function to find the lift response to a change in the angle of attack, while the compressibility effect is taken into account via a correction introduced by Leishman and Beddoes.¹⁶ Once the aerodynamic characteristics are obtained, the MST is applied to acquire the circulation strength at control points. It was shown that seven points are enough to correctly represent the loading without any sharp discontinuities.¹⁷ The interaction between the main wing and the horizontal tailplane is modelled through the state-space downwash model.

The modelling of the fuselage-wing interaction and aerodynamics of the flexible fuselage and engines is based on the ESDU methodologies. The fuselage is modelled as a slender axi-symmetric body of revolution.^{18–20} An engine nacelle is modelled as an annular aerofoil.²¹

II.B. Structural model

The structural model in the CA²LM framework is based on the small deformation theory, thus assuming linear behaviour of deformations. The aerodynamic and gravitational forces and moments are translated into deformations of the airframe in the modal domain. Hence, the modeshape data of the airframe (the modeshape matrix, modeshape frequencies and masses) are acquired using the Nastran solver SOL103. As the aerodynamic and structural models are coupled to provide the aeroelastic analysis framework, the modal deformations are translated to aerodynamic frame deformations by simple interpolation.

II.C. Equations of motion

Equations of motion (EoM) translate the aerodynamic and gravitational forces and moments to flight dynamics parameters. However, as the CA²LM framework is intended to model flexible aeroplanes, the classical EoM have been amended with terms, which take into account the flexibility of aircraft by adding relative accelerations, velocities and displacements of structural points with respect to the body axes centre. These additional terms introduce algebraic loops. It will be discussed later, that an algebraic loop introduces significant issues within MATLAB[®]/Simulink[®] models. A different formulation of EoM might be used to avoid an algebraic loop. However, their re-formulation in integral form is purely mathematical problem, which is out of scope in the current stage of the project.

II.D. Critical discussion of the CA²LM framework

Current implementation of the CA²LM framework provides both advantages and disadvantages.

Low fidelity aerodynamic models along with the linear structural model are fast to compute and provide results, precise enough to rapidly evaluate concept aircraft configurations in the early stages of development.

However, large wing deformations cannot be modelled precisely as they possess non-linear behaviour. Hence, a non-linear structural dynamics solver should be implemented. Moreover, aerodynamic models limit the range of angle of attack to $\pm 15^\circ$ as they are based on pre-computed look-up tables and the stall is not modelled.

The current version of the CA²LM framework models only primary control surfaces, i.e. ailerons, an elevator and a rudder. Recently, spoilers have been added to the wing aerodynamic model for load alleviation purposes. Thus, only cruise conditions might be analysed realistically within the current framework. An introduction of secondary surfaces, such as flaps and slats, would broaden the applicability range of the framework to climb and descent/initial approach flight stages.

The MATLAB[®]/Simulink[®] environment provides easy and fast access to the models within the framework. However, as mentioned earlier, the algebraic loops induce significant issues for large and complex models. Thus, ‘unit delay’ blocks have been introduced to break the algebraic loops between the aerodynamic and structural models and within the EoM due to additional terms due to flexibility. The ‘unit delay’ block, by its definition, uses states from the previous time step, hence introducing inaccuracies in the simulation. However, this solution helps not only to break algebraic loops, but to increase the computational speed and to integrate the model into a real-time simulator (an integration procedure is discussed later in the paper).

Nevertheless the previous solution for algebraic loops improves the computational pace, the framework still lacks computational speed. However, this issue is acceptable knowing the complexity of the framework due to the aeroelastic modelling. One of possible ways to increase its computational speed is to use alternative hardware, e.g. field-programmable gate array (FPGA) circuits. However, this requires re-programming of the entire framework into the FPGA compatible format. Additionally, the framework is intended to be used by engineers on their desktop machines, thus acquiring additional hardware to use the framework would reduce its versatility. Hence, it was decided to mitigate the issue through the use of a dedicated real-time simulation hardware - the Speedgoat real-time target machine, the use of which is explained in Section IV.

III. Engineering Flight Simulator

The target simulator for the integration of the CA²LM framework is the EFS500 simulator within the Aerospace Integration Research Centre (AIRC) at Cranfield University. The EFS500 provides a platform for the pilot to interface with the CA²LM framework and constitutes another important part of a new simulation framework. It is an engineering flight simulator designed to demonstrate aircraft stability and control along with aircraft performance in real time, which is achieved by:

1. Modelling only the rigid body dynamics excluding any structural dynamics.
2. Aerodynamic and flight dynamic coefficients being modelled by simple empirical formulations.

Its hardware architecture and software are presented in the following subsections.

III.A. Hardware

The EFS500 simulator is comprised of the following elements:

- *Cockpit environment*, which provides a generic single seat cockpit layout with a central stick, sidestick, rudder pedals and throttle lever. The flap selector, gear switch and other additional switches are also included in the cockpit, which is shown in Fig. 1(a).
- *Artificial loading system*, which allows force feedback response to the central stick and rudder pedals.
- *Computer system*, which consists of 8 computers that provide flight dynamics model, cockpit displays, avionics, instructor operating station and image generation capability.
- *Sound generation system*, which generates environmental and warning sounds.
- *Image generation system*, which is comprised of 3 projectors and a spherical screen, which provides a field of view of $140^\circ \times 40^\circ$ as shown in Fig. 1(b).



(a) Cockpit



(b) Overall setup



(c) EFS500 flight simulator during a testing campaign

Figure 1. EFS 500 flight simulator environment

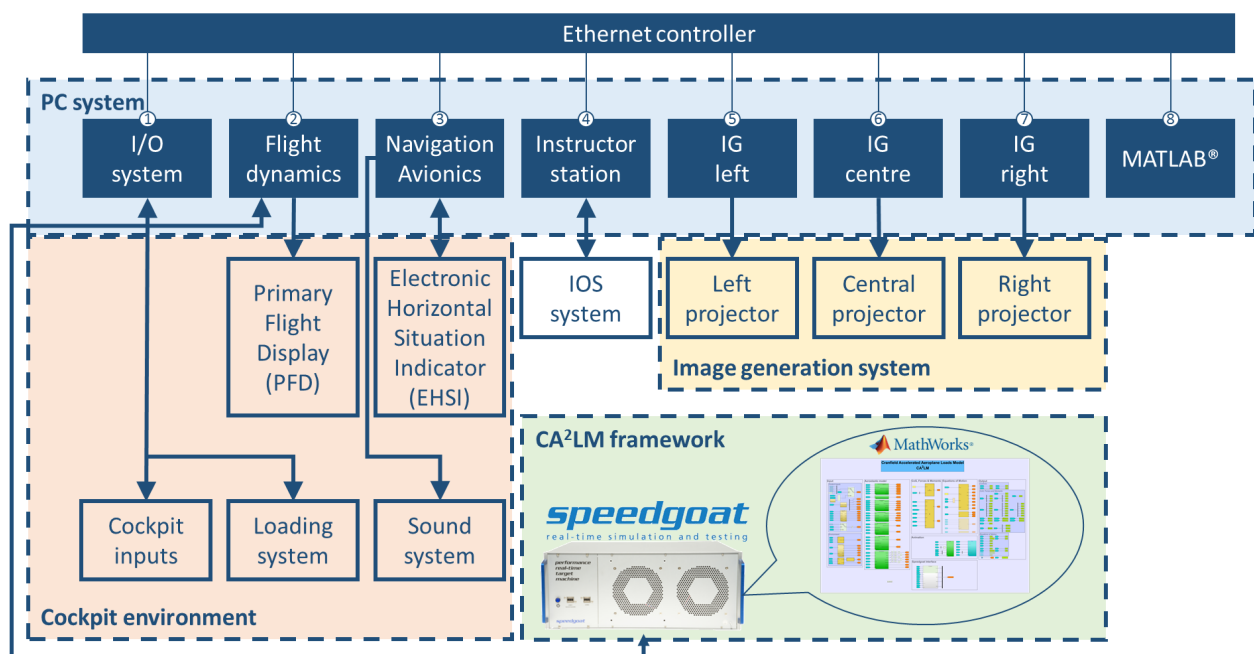


Figure 2. EFS500 simulator network

Computers in the simulator are connected into a segregated Ethernet network and communicate broadcasting information from each node sequentially. The communication sequence is controlled by the I/O system node operating at a frequency of 50Hz. The overall system architecture is shown in Fig. 2.

Each of the eight computers performs the following tasks:

1. The I/O system computer is based on a Raspberry Pi[®] system to interface between the control inputs of the simulator hardware (i.e. control stick, right-hand stick, pedals, throttle) and the computer system. It has 32 analogue/digital input channels and 4 analogue/digital output channels. Moreover, this computer synchronises the computer system to broadcast packets of data at a frequency of 50 Hz.
2. The flight dynamics computer provides equations of motion, an aerodynamic model, engine models, flight control laws, dynamics of targets and the weather model. Moreover, it renders the primary flight display (PFD) as seen in Fig. 3(a).
3. The navigation and avionics computer computes navigation equations for the aircraft sensors, manages radio and flight control unit (FCU) panels and generates aircraft sounds. Moreover, it renders the electronic horizontal-situation indicator (EHSI) as seen in Fig. 3(b).
4. The instructor operating station provides an interface of commanding different flight conditions and data recording via the touch screen GUI. It provides a capability of rapid repositioning of an aircraft on the map, plots simulator output data, allows failures of different aircraft systems, controls autopilot and flight plan.
- 5-7. The image generation (IG) computers are equipped with high performance nVidia[®] graphics cards. They provide an OpenFlight[®] visual database and render it in response to the data received from other computers, i.e. aircraft position, its attitude, weather conditions, time of day. Moreover, the head-up display may be displayed by IG computers.
8. The MATLAB[®] computer provides an interface for the rapid implementation and testing of control laws.

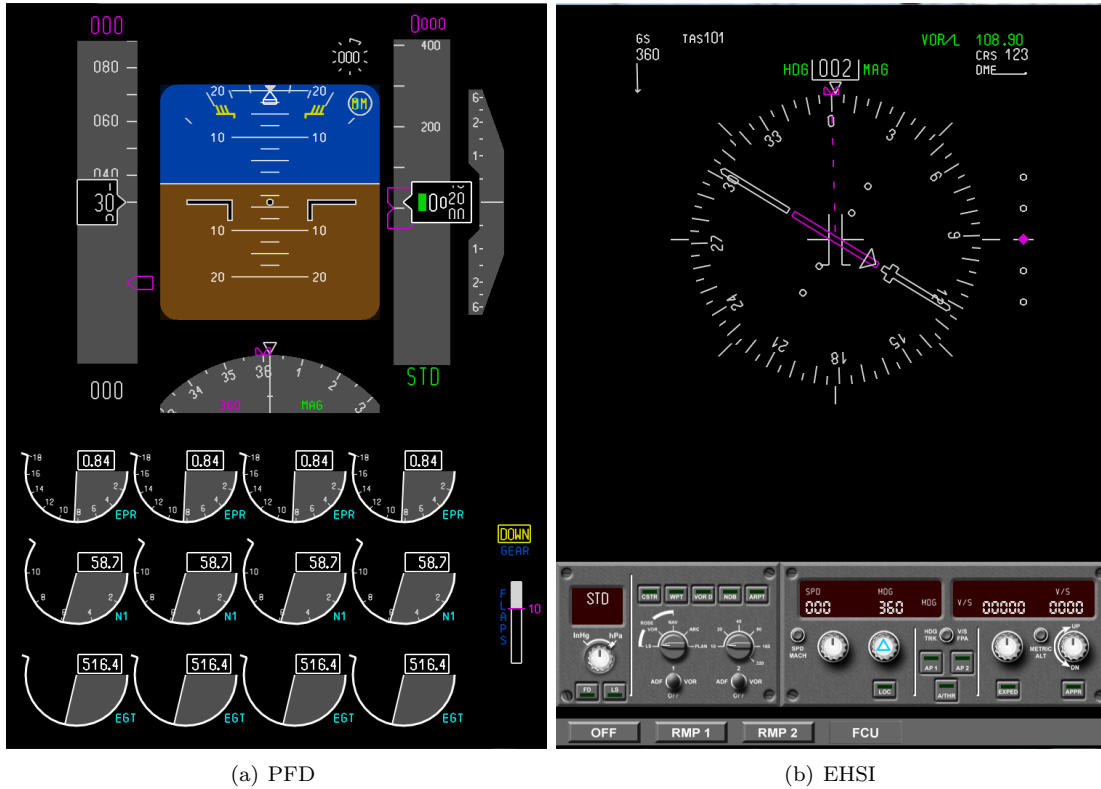


Figure 3. PFD and EHSI displays in EFS500 flight simulator cockpit

All computers are connected into a local network by the 16-channels Ethernet switch. As the flight dynamics software runs at 50 Hz, so every other computer has to run at the same frequency. It is very important not to exceed 20 ms frame time when doing computations, thus the margin is included in the computations of each computer.

III.B. Software

The simulator runs custom coded software developed mostly in the C programming language. The code is distributed within the computers, allowing rapid modifications of the code and recompilation of the applications. Nevertheless that the most of simulator's software is written in the C programming language, image generation computers use the C++ programming language to interface with the OpenSceneGraph graphics library. All the nodes use the 'gcc' compiler for both C and C++ applications.

The simulator software is intended to be as modular as possible and the modification of the aircraft model only affects particular nodes. Details of each node function are presented in Table 1. Some of the nodes, like the instructor operating station, the I/O system or the image generation node, are not affected by changes in the aircraft model.

Table 1. Computer node function and aircraft model dependency

Node	Name	Function	Aircraft dependency
1	I/O system	Acquire data from cockpit and enable load system	No
2	Flight dynamics	Compute flight dynamics and render primary flight display	Yes
3	Navigation & avionics	Compute avionics and render secondary flight display	Yes
4	Instructor station	Run instructor software enabling simulation control	No
5	IG left	Render left area of the 3D environment	No
6	IG centre	Render central area of the 3D environment	No
7	IG right	Render right area of the 3D environment	No
8	MATLAB	Rapid implementation and testing of flight control laws	N/A

Only one application is executed per node as part of the original network design of the simulator. The flight dynamics computer (Node 2) process flow is shown in Fig. 4.

It is important to notice that the application is idle while the network data is being transferred. This architecture of the process flow relies on the network transaction time, which is supposed to be notably smaller than the total available slot time, i.e. less than 20 ms. In case smaller time steps or higher number of nodes in the architecture are required, the time spent in network transactions may significantly reduce the time available for calculations.

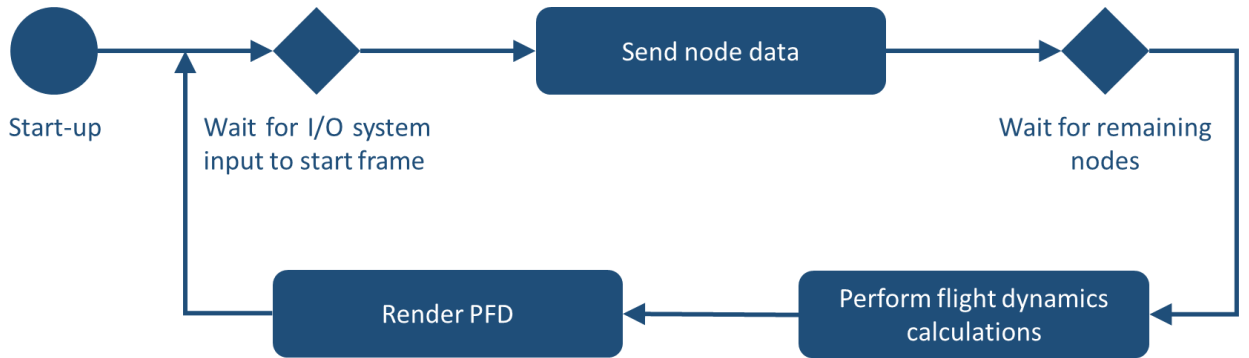


Figure 4. Flight dynamics computer (Node 2) process flow

IV. Integrating the CA²LM Framework into the EFS500 Flight Simulator

To allow the interaction between the CA²LM framework, developed in the MATLAB[®]/Simulink[®] environment, and the EFS500 flight simulator, developed in C/C++ programming languages, both architectures

must be compatible. There are two main areas that need to be compatible between the CA²LM framework and the EFS500 flight simulator:

1. *Timing*, as both systems need to run in real time in order to perform the pilot-in-the-loop simulations. The EFS500 flight simulator is running in real time by default, but the CA²LM framework is much slower than the real time. To increase the simulation speed, the Speedgoat real-time target machine is used.
2. *Data interfaces*, as the information exchange between the two systems must be compatible. The Simulink Real-Time software needs to be capable of sending enough information for the EFS500 flight simulator to work, while the simulator should feed the the real-time device with the expected control inputs.

The simulation timing requirement imposes the need for the CA²LM framework to run in real time. At the current level of implemented aircraft model complexity, the CA²LM framework is capable of performing 100 iterations per second on a standard i7-4770 desktop computer. Considering that the time-step used in the framework is 0.001 s (1000 Hz), the CA²LM framework needs to be sped-up by a factor of 10.

In order to obtain the required performance improvement, it has been decided to use the Speedgoat real-time target machine, which specifications are described in Table 2.

Table 2. Speedgoat real-time target machine specifications

Element	Model
CPU	Intel i7-6700K
RAM	8192MB
Storage	60GB SSD
Ethernet	2 + 1 Gigabit
OS	FreeDOS / Simulink Real-Time™ kernel

The implementation of the CA²LM framework in the Speedgoat device requires some changes in the model. Firstly, the data interface between the EFS500 and the CA²LM framework needs to be compatible and compliant with the real-time requirement. An option would be to synchronise both systems, with the CA²LM framework executing the simulation time step when the simulator network has finished the data transaction. Since the CA²LM framework, with the current aircraft model implemented, executes the simulation at 1000 Hz and the EFS500 runs at 50 Hz, a synchronised system is not suitable. An asynchronous system is used instead, with both the simulator system and the CA²LM framework running freely and exchanging information asynchronously. The external aeroelastic framework integration is therefore independent of its simulation frequency.

Since the PFD computer was originally responsible for flight dynamics calculations, it was decided that it will also be responsible for interfacing with the CA²LM framework (see Fig. 2). The primary data interface to the Speedgoat real-time machine is the Ethernet port. The architecture of the EFS500 network requires that no other data, except for the simulation data, is sent using the simulator network. Therefore, communication between the PFD computer and the real-time machine is done using a secondary Ethernet network independent of the main simulator network. The resultant architecture, presented in Fig. 5, comprises of two separate networks with the PFD computer acting as the gateway between them. To allow this architecture, a secondary Ethernet port has been installed in the PFD computer.

Communication in the secondary network uses user datagram protocol (UDP) packets. In order to enable full control of the simulation, three packets have been defined:

- *Flight dynamics output packet*, which is used for the Speedgoat machine to send the most recent set of computed data to the flight model computer. This contains position, attitude, translational and angular velocities of an aircraft along with flight conditions data. The information could be potentially sent at 1000 Hz, but it is sub-sampled and sent at 50 Hz, matching the frequency of the EFS500 simulator.
- *Simulation data packet*, which contains the simulation information that is generated in other simulator nodes. It includes control status, simulation status and ground altitude, which is computed in the Node 2 computer.

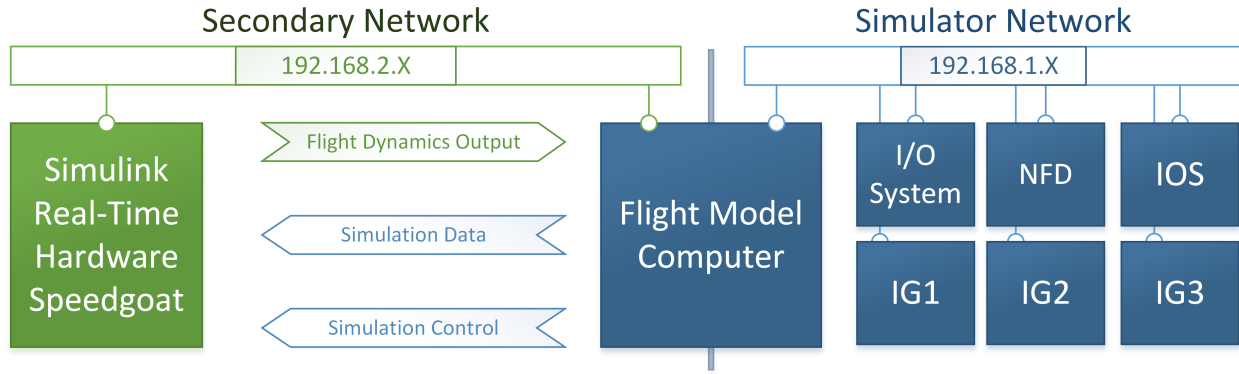


Figure 5. Interface between the PFD computer and the Speedgoat real-time machine

- *Simulation control packet*, which contains simulation modifiers, such as simulation start/stop flag, aircraft repositioning flag or simulation restore flag.

UDP interfaces in the real-time machine and the PFD computer are configured as non-blocking, always using the most recent value to allow systems to run asynchronously.

The Speedgoat real-time target machine requires the CA²LM framework to be loaded into the target machine at each simulation. Two changes to the CA²LM framework have been made to allow the compilation of the framework for the real-time target machine. The changes are following:

1. *Disabling (or ‘commenting out’ in MATLAB[®]/Simulink[®] terms) Level-2 S-Functions*, which are used to generate visualisation of the airframe flight and structural dynamics during the simulation. Level-2 S-Functions can be compiled into real-time machines as long as C or Fortran code is provided, but the visualisation code within the CA²LM framework is not available in those languages. Since the visualisation functions are redundant for the real-time simulations, the functions are simply disabled when targeting the Speedgoat real-time machine.
2. *Break of algebraic loops*, which introduce issues additionally to the previously discussed ones. Simulink[®] can solve algebraic loops in a discretised model when running in normal mode, but they need to be removed before compiling the model for a real-time machine. Algebraic loops can also slow down the simulation. Hence, the ‘unit delay’ blocks were added to break algebraic loops.

After making these modifications and specifying the required configuration in the ‘Model Configuration Dialogue’ window, the model is compiled and loaded into the real-time device. The report from the device indicates that the computation time for each time step is 0.7 ms, or approximately 70% of the available time. Hence, as the model runs in real time it can be integrated into the EFS500 simulator.

To comply with the UDP interface, Speedgoat real-time machine network configuration blocks have been added along with the following 3 UDP interfaces:

- *UDP receiver for simulator data*, connected to a packet decoder (‘Byte Unpack’) to provide simulation parameters.
- *UDP receiver for simulator control*, connected to a packet decoder (‘Byte Unpack’) to extract the simulation commands such as stop or reposition.
- *UDP sender for output data*, preceded by a ‘Byte Pack’ element to send the output of the simulation to the PFD computer.

To ensure smooth integration the number of modifications has been kept to a minimum. Following the description in Table 1, the only nodes that should be modified regarding the model are Nodes 2 and 3, featuring flight dynamics, avionics and screens rendering. However, only the flight dynamics code running in the Node 2 computer has been modified, encapsulating all the changes in the simulator into one single computer.

Additionally to installing an Ethernet port to communicate to the Speedgoat real-time target machine, the following changes have been performed on the Node 2 computer code to allow the integration of the CA²LM framework:

1. The original flight dynamics code has been removed from the EFS500 simulator.
2. The bypass has been implemented to allow the data from the CA²LM framework to be used instead of the original flight dynamics code.

V. Example of Handling Qualities Testing Campaign

To show the application example of the new simulation framework the HARTEN (High Aspect Ratio Technology ENabler) aircraft model was tested for its controllability. HARTEN is the research model developed by Cranfield University, the University of Bristol, and Airbus, during the Agile Wing Integration (AWI) project to investigate the feasibility of using high aspect ratio wings for commercial transport aircraft. Cranfield University has been tasked to evaluate the handling qualities of the HARTEN aircraft, this being the initial reason for developing the new flight simulation framework.

HARTEN is the high-aspect-ratio unswept wing aircraft. The configuration of the HARTEN model is provided in Fig. 6.

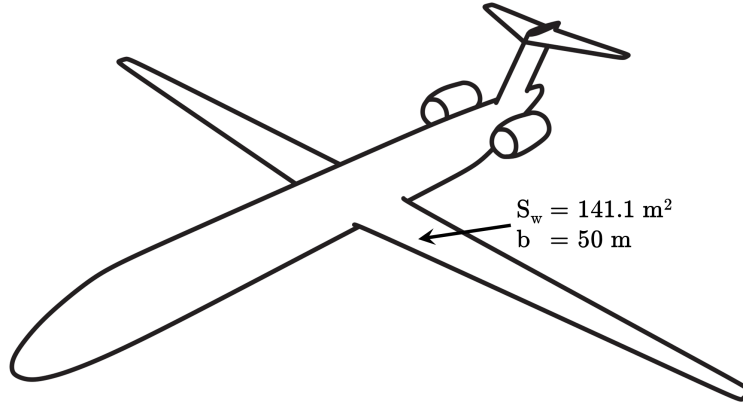


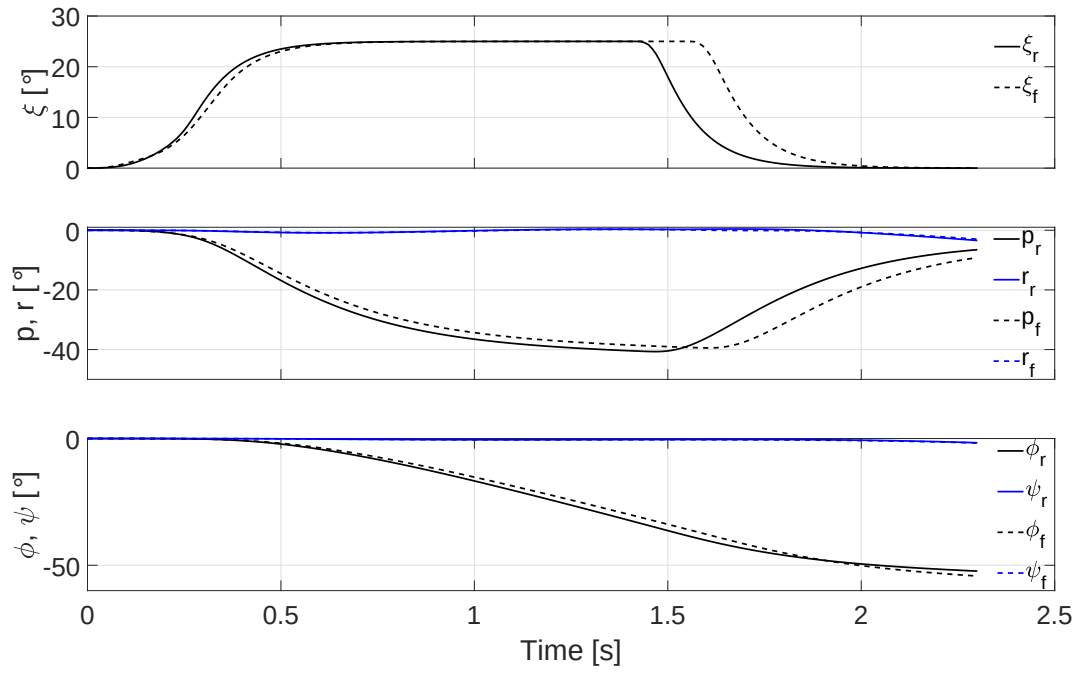
Figure 6. HARTEN conceptual aircraft model definition

Two airframe flexibility cases were tested for the example. In both cases the fuselage and the empennage were modelled as rigid structures. However, the wing was modelled as a rigid structure by setting Young's modulus value to 8×10^{10} and as a flexible structure by setting E value to 8×10^9 . These two versions will be referred to as rigid and flexible cases.

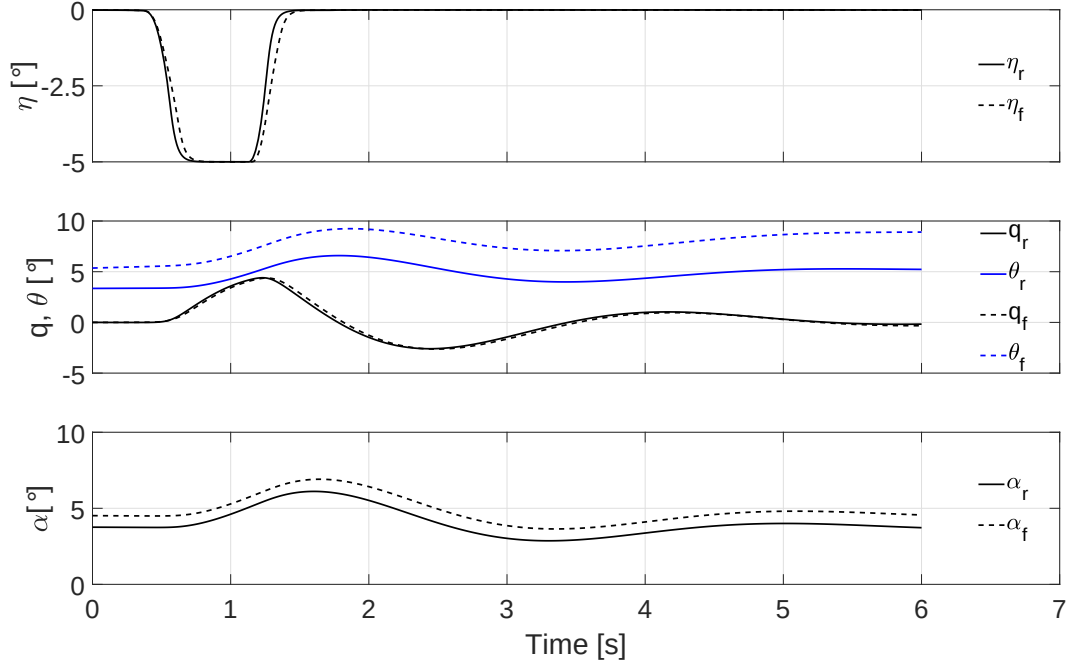
The $\pm 30^\circ$ bank angle roll manoeuvre has been carried out to investigate lateral controllability of the model and structural flexibility influence to it. The short period pitching oscillation has been analysed to evaluate the longitudinal stability of the model. Results for both testing cases are presented in Fig. 7.

The roll dynamics test evaluates the aircraft response to an aileron input. The command to maximum aileron deflection of $\xi = 25^\circ$ was provided for about 1.5 s. The control input dynamics and the HARTEN aircraft response is provided in Fig. 7(a). The coupling between the roll p and yaw r dynamics is clearly seen in the figure, i.e. negative roll moment induces negative yaw moment, thus the aircraft banks and turns to the left as a result of positive aileron input. However, the adverse yaw effect is seen between 0.5 s and 1.5 s as the magnitude of the yaw rate reduces instead of increasing with the increasing roll rate. This is the common feature of the high aspect ratio wing configurations. Lag of the lateral/directional dynamics due to the airframe flexibility is minimal.

Another test case carried out for the HARTEN aircraft model evaluates its short period pitching oscillation (SPPO) mode. The command to elevator deflection of $\eta = -5^\circ$ was provided for about 0.5 s. The control input dynamics and the HARTEN aircraft response is provided in Fig. 7(b). The first insight shows that the pitch angle θ and the angle of attack α for rigid and flexible bodies are different. Both angles for the flexible wing case are higher due to reduced effective wing span.



(a) Roll dynamics



(b) SPPO mode dynamics

Figure 7. HARTEN lateral and longitudinal dynamics

VI. Conclusions and Further Work

Engineering flight simulators provide the capability of pilot-in-the-loop testing of novel aircraft concepts at the very early stages of design. One such simulator has been developed at Cranfield University by integrating the CA²LM framework with the EFS500 real-time flight simulator. The integration process has been discussed in the paper. One of the biggest issues was a lack of real-time simulation capability using the CA²LM framework. This has been mitigated by the use of the Speedgoat real-time target machine. However, the presence of algebraic loops, which were treated by the MATLAB[®]/Simulink[®] software when running the CA²LM framework independently, was noted to be one of the biggest issues when transferring the CA²LM framework to the Speedgoat real-time target machine. The issue was solved by adding ‘unit delay’ blocks to the signals in the algebraic loops. The synchronisation issue between two simulators has been solved by the adoption of an asynchronous system.

Finally, the new flight simulation framework has been tested by investigating lateral and longitudinal handling qualities of the HARTEN conceptual aircraft model. The flexibility effect on the flight dynamic parameters has been clearly seen for the pitch angle and the angle of attack. However, for the dynamics in roll only slight lags have been noticed.

Regarding the development of the new flight simulation framework, multiple limitations and further work elements have been identified:

1. A wider flight envelope should be provided within the CA²LM framework by increasing available range of angles of attack.
2. A software development kit (SDK) should be developed for easier and clearer implementation and testing of new capabilities within the new flight simulation framework.
3. The CA²LM framework should include a non-linear structural model for modelling of high wing deformations.

Acknowledgments

The authors wish to thank Airbus, the Aerospace Technology Institute and Innovate UK for supporting the research through the Agile Wing Integration (AWI) project.

References

- ¹Seitz, A., Kruse, M., Wunderlich, T., Bold, J., and Heinrich, L., “The DLR Project LamAiR: Design of a NLF Forward Swept Wing for Short and Medium Range Transport Application,” American Institute of Aeronautics and Astronautics, June 2011.
- ²Saeed, T. I. and Graham, W. R., “Design Study for a Laminar-Flying-Wing Aircraft,” *Journal of Aircraft*, Vol. 52, No. 5, Sept. 2015, pp. 1373–1385.
- ³Uranga, A., Drela, M., Greitzer, E. M., Hall, D. K., Titchener, N. A., Lieu, M. K., Siu, N. M., Casses, C., Huang, A. C., Gatlin, G. M., and Hannon, J. A., “Boundary Layer Ingestion Benefit of the D8 Transport Aircraft,” *AIAA Journal*, Vol. 55, No. 11, Nov. 2017, pp. 3693–3708.
- ⁴“Flightpath 2050: Europe’s Vision for Aviation,” Tech. rep., European Commission, Luxembourg, 2011.
- ⁵Tollefson, J., “UN Agency Proposes Greenhouse-Gas Standard for Aircraft,” *Nature*, Vol. 530, No. 7590, Feb. 2016, pp. 266–266.
- ⁶Pitera, D. M., DeHaan, M., Brown, D., Kawai, R. T., Hollowell, S., Camacho, P., Bruns, D., and Rawden, B. K., “Blended Wing Body Concept Development with Open Rotor Engine Integration,” Tech. Rep. NASA/CR-2011-217303, NASA Langley Research Center, Hampton, VA, Jan. 2011.
- ⁷Cooper, J., Lowenberg, M., Lone, M. M., Garry, K., Cooke, A., and Coetzee, E., “High Aspect Ratio Technology Enablers - HARTEN. Research Proposal,” Feb. 2014.
- ⁸Bradley, M. K., Droney, C. K., and Allen, T. J., “Subsonic Ultra Green Aircraft Research Phase II: Volume I – Truss Braced Wing Design Exploration,” Tech. Rep. NASA/CR-2015-218704/Volume I, NASA, Washington, DC, Jan. 2015.
- ⁹Allerton, D., *Principles of Flight Simulation*, Wiley, Chichester, UK, 2009.
- ¹⁰Andrews, S. P. and Cooke, A. K., “An Aeroelastic Flexible Wing Model for Aircraft Simulation,” American Institute of Aeronautics and Astronautics, Orlando, FL, Jan. 2010.
- ¹¹Andrews, S. P., *Modelling and Simulation of Flexible Aircraft: Handling Qualities with Active Load Control*, Phd thesis, Cranfield University, Cranfield, UK, March 2011.
- ¹²Lone, M. M., *Pilot Modelling for Airframe Loads Analysis*, Phd thesis, Cranfield University, Cranfield, UK, 2013.
- ¹³Portapas, V., Cooke, A., and Lone, M. M., “Modelling Framework for Flight Dynamics of Flexible Aircraft,” *Aviation*, Vol. 20, No. 4, Oct. 2016, pp. 173–182.

- ¹⁴Portapas, V., Yusuf, S. Y., Lone, M. M., and Coetzee, E., “Modelling Framework for Handling Qualities Analysis of Flexible Aircraft,” American Institute of Aeronautics and Astronautics, Grapevine, TX, Jan. 2017.
- ¹⁵Leishman, J. G. and Nguyen, K. Q., “State-Space Representation of Unsteady Airfoil Behavior,” *AIAA Journal*, Vol. 28, No. 5, May 1990, pp. 836–844.
- ¹⁶Leishman, J. G. and Beddoes, T. S., “A Generalised Model for Airfoil Unsteady Aerodynamic Behaviour and Dynamic Stall Using the Indicial Method,” Washington, DC, June 1986, pp. 243–265.
- ¹⁷DeYoung, J. and Harper, C. W., “Theoretical Symmetric Span Loading at Subsonic Speeds for Wings Having Arbitrary Plan Form,” Tech. Rep. NACA-TR-921, NACA Ames Aeronautical Lab., Moffett Field, CA, Jan. 1948.
- ¹⁸“Normal Force and Pitching Moment of Conical Boat-Tails,” Tech. Rep. ESDU 87033, ESDU, July 1992.
- ¹⁹“Normal-Force-Curve and Pitching-Moment-Curve Slopes of Forebody-Cylinder Combinations at Zero Angle of Attack for Mach Numbers up to 5,” Tech. Rep. ESDU 89008, ESDU, Dec. 1990.
- ²⁰“Normal Force, Pitching Moment and Side Force of Forebody-Cylinder Combinations for Angles of Attack up to 90 Degrees and Mach Numbers up to 5,” Tech. Rep. ESDU 89014, ESDU, Sept. 2004.
- ²¹“Aerodynamic Centre of Wing-Fuselage-Nacelle Combinations: Effect of Wing-Pylon Mounted Nacelles,” Tech. Rep. ESDU 77012, ESDU, Nov. 2013.



DSSPACE

<https://dspace.org/>

Pilot-in-the-loop flight simulation of flexible aircraft in Matlab / Simulink: Implementation and coding peculiarities

Lopez Matos, Guillermo E.; Portapas, Vilius; Dussart, Gaétan X.; Lone, Mohammad M.; Coetzee, Etienne

2018-01-13

AIAA

<http://dx.doi.org/10.2514/6.2018-0426>

Lopez Matos GE, Portapas V, Dussart GX, et al., (2018) Pilot-in-the-loop flight simulation of flexible aircraft in MATLAB/Simulink: implementation and coding peculiarities. In: Proceedings of AIAA 2018 Modeling and Simulation Technologies Conference, 8-12 January 2018, Kissimmee, Florida, USA, Paper number AIAA 2018-0426