A Co-Simulation Framework for Aircraft Ground Dynamics and Lateral Control Integration

Valerian-Marian SANDU[1], Tudorel-Petronel AFILIPOAE[1], Ioan Petru UDREA[1], Adrian TOADER*,[1]

*Corresponding author
[1]INCAS – National Institute for Aerospace Research “Elie Carafoli”,
B-dul Iuliu Maniu 220, Bucharest 061126, Romania,
sandu.valerian@incas.ro, afilipoae.tudorel@incas.ro, udrea.petru@incas.ro, toader.adrian@incas.ro

DOI: 10.13111/2066-8201.2025.17.4.17

Received: 24 October 2025/ Accepted: 11 November 2025/ Published: December 2025

Abstract: *This research introduces a simulation framework for aircraft ground roll, integrating a dynamic model developed in Simcenter Amesim with a control system implemented in MATLAB/Simulink through the Functional Mock-up Interface (FMI). The study aims to provide a robust environment for analysing and validating advanced ground control strategies under realistic runway roll conditions. The aircraft model, based on a 6DOF tricycle configuration, includes steering and differential braking features adapted from a validated Amesim taxi model, using representative data of a commercial turboprop aircraft. The framework enables bidirectional communication between Amesim and MATLAB through FMI and Co-Simulation, allowing performance evaluation of the control system. The controller employs a structured H-Infinity approach combined with a control allocator to coordinate rudder, nose wheel, and differential braking actions, ensuring runway centerline tracking and stability even in the presence of crosswind and for different runway conditions. The FMI integration process includes interface remapping, FMI export from Simulink, and closed-loop coupling with the Amesim model. Additional simulations were performed using a reverse configuration, with Amesim running as a co-simulation module within MATLAB via an S-function interface, to validate interoperability and numerical consistency. Comparative analysis between FMI and Co-Simulation confirm consistent data exchange and equivalent dynamic responses across platforms. The developed framework demonstrates the feasibility, flexibility, and accuracy of hybrid Amesim–MATLAB environments for studying complex aircraft ground dynamics configurations and provides a foundation for future work.*

Key Words: *Aircraft Ground roll, Functional Mock-up Interface (FMI), Co-Simulation, control system, H-infinity*

1. INTRODUCTION

Aircraft ground dynamics represent a critical phase in the flight operation cycle, encompassing the motion of the aircraft during taxiing, take-off, and landing. The accurate modelling and control of ground operations are essential for safety, comfort, and efficiency. Traditionally, control systems for ground manoeuvres are developed and tested within standalone simulation environments, limiting the ability to assess coupled effects between mechanical, aerodynamic, and control subsystems. Recent advances in system-simulation environments such as

Simcenter Amesim [1] and MATLAB/Simulink [2] facilitate the coupling of multi-domain models via standardized interfaces such as the Functional Mock-up Interface (FMI) [3]. The FMI standard provides a framework for interoperability among modelling tools by enabling model exchange and co-simulation through self-contained units known as Functional Mock-up Units (FMUs). This capability is of growing importance in the design of advanced control systems, which require interaction between mechanical, electrical, thermal and control domains.

This paper presents a co-simulation framework for aircraft ground dynamics and control integration, combining a 6DOF dynamic ground model in Simcenter Amesim with a structured H-Infinity controller designed in MATLAB/Simulink. The study focuses on the implementation, interoperability validation, and performance evaluation of the co-simulation approach using FMI. The development of the framework involves the encapsulation and integration of models constructed across multiple platforms, with the objective of enabling future analysis of aircraft ground roll and taxiing dynamics, while ensuring modularity and facilitating cross-platform analyses.

2. METHODOLOGY

2.1 Framework overview

The proposed framework in the paper integrates two major components: a 6DOF ground dynamics model developed in Simcenter Amesim and a lateral control system implemented in MATLAB/Simulink. Both environments communicate via the FMI 2.0 Co-Simulation standard, Fig. 1, enabling signal exchange and synchronized solver execution during simulation.

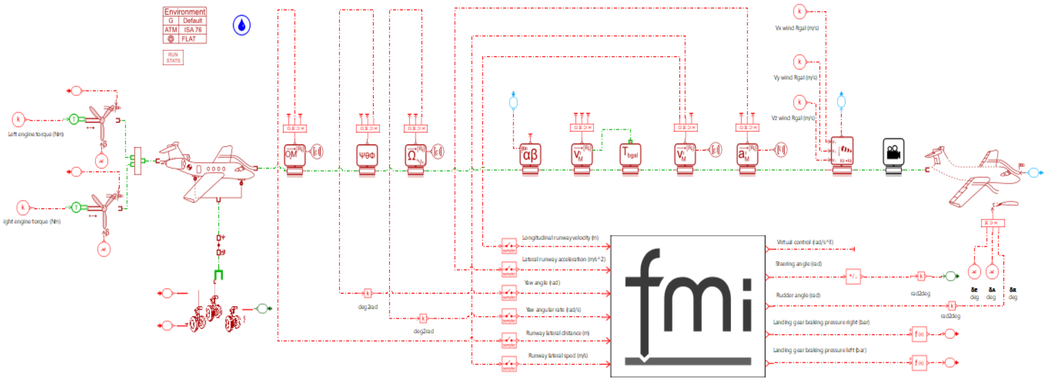

Fig. 1 Ground dynamics model with FMI controller

The workflow adopted in this study follows a structured procedure aimed to enable cross-platform controller integration and evaluation. Initially, the control algorithm is developed and validated within the MATLAB/Simulink environment to ensure functional correctness and stability. Upon successful validation, the controller is exported as a Functional Mock-up Unit (FMU) in compliance with the Functional Mock-up Interface (FMI) standard. The generated FMI is then imported into Simcenter Amesim, where the corresponding input–output interfaces are systematically mapped to establish consistent signal exchange between the control and plant models. Subsequently, closed-loop co-simulation runs are performed within the integrated environment to assess the dynamic performance of the system and to verify the correct interaction between the mechanical and control subsystems.

2.2 Aircraft ground dynamics model

The aircraft model employs a 6-DOF tricycle landing gear configuration [4], suitable for studying ground handling during take-off and landing. It integrates wheel dynamics, steering, and braking subsystems, with nonlinear contact forces computed using Amesim's tire model. Developed from an existing taxiing model, it utilizes blocks from the *Aerospace and Marines* and *Vehicle Dynamics* Amesim libraries [1], which allow separate subsystem modeling and integration into a high-fidelity simulation. Fig. 2 illustrates the connection between the landing gear and the aircraft dynamics block (solving the 6-DOF nonlinear motion equations), while Fig. 3 shows the *Vehicle Dynamics* library blocks modelling tire–ground interaction. Ground reaction forces from each gear (left, right, and nose) are summed and transmitted to the aircraft dynamics module.

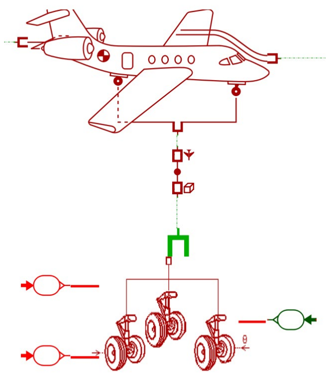

Fig. 2 Landing gear aircraft interaction

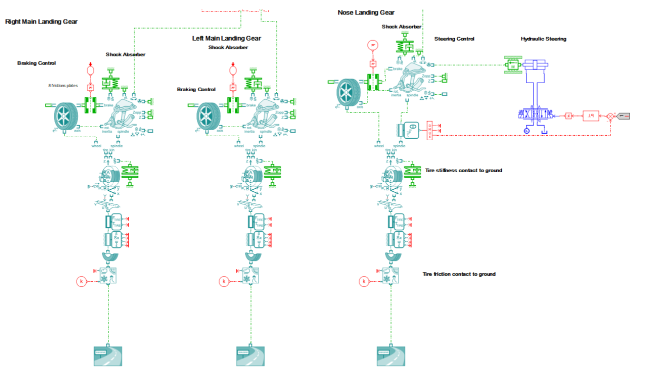

Fig. 3 Landing subsystem

The landing gear model is structured from the ground up, beginning with a flat-ground module that interfaces with an adherence generator, computing tire–road contact forces. The subsequent tire belt model determines camber, side-slip, and longitudinal slip angles, while the tire element incorporates spring and damper characteristics. The hub carrier model connects the wheel (including its inertia) and shock absorbers to the aircraft structure, with a frictional torque element enabling differential braking between the left and right main gears. The taxiing model was adapted to support this functionality (differential braking) without altering landing gear parameters. Furthermore, the nose gear integrates a hydraulic actuator and proportional controller for steering angle tracking. The subsystem enables precise ground run model behaviour through adjustable grip coefficients for various runway conditions (dry, wet, or icy).

2.3 Control System Design and Implementation

The closed-loop control system is designed to minimize aircraft lateral deviation from the runway centerline by coordinating the rudder, nose wheel, and differential braking actions. Longitudinal motion (position and velocity) is controlled in open loop by the pilot through throttle and total braking pressure.

The architecture includes a structured H-Infinity controller generating a virtual control signal (yaw rate derivative torque) and a control allocator that distributes it among the aircraft effectors using a least-squares algorithm. A nonlinear aircraft ground model was developed in Simulink based on [4], assuming a tricycle configuration, planar rigid-body motion, small sideslip and nose-wheel angles, and linearized aerodynamics. The model includes first-order actuator with deflection and rate saturations and a low pass sensors dynamics response with noise, and a validated semi-empirical tire–ground contact model.

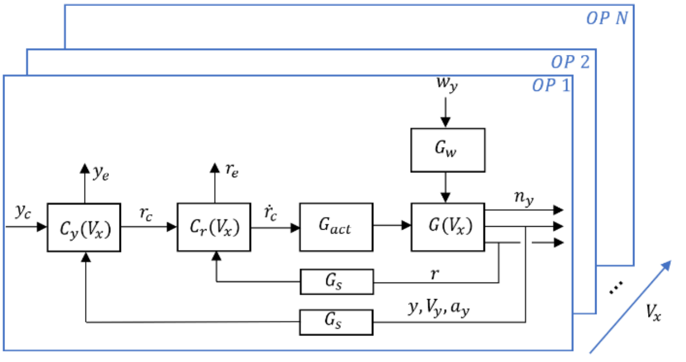

Fig. 4 Closed loop system

Moreover, the nonlinear model has been linearized in order to tune the H-Infinity controller gains using a structured H-Infinity optimization method (*systune*) from MATLAB [2]. The structure of the controller is presented in Fig. 4.

The closed loop control diagram is comprised of:

- G – linearized aircraft lateral dynamics with V_x as parameter
- G_{act}, G_s - actuators and sensors dynamics (first order)
- G_w - wind bandwidth limitation (first order)
- C_y - lateral position controller, fixed structure
- C_r - yaw rate controller, fixed structure

While the close loop signals are:

- y, V_y, a_y, n_y, r- realized lateral position, velocity, acceleration, load factor and yaw rate
- y_c, W_y - commanded position and lateral wind
- y_e, r_e - position and yaw rate errors
- $r_c, \dot{r}_c$ - commanded yaw rate and yaw rate derivative (virtual control signal)

Performance requirements on, peak tracking error and overshoot, maximum lateral load factor under position command and crosswind disturbance, tracking error and rejection time under crosswind disturbance were defined as inputs to the *systune* such this function to output a linear controller that meets these requirements.

$$K_i(V_x) = a_0 + a_1 V_x + a_2 V_x^2 \quad (1)$$

Robustness of the H – Infinity controller has been also taking into account in the design by setting uncertainties on the aircraft mass, inertia and geometry, aerodynamic coefficients, air density, contact forces, and on velocity along the runway (V_x), to account headwind/tailwind.

A Control allocation [4] approach is applied to distribute the H-infinity virtual control $v(t)$ to the aircraft effectors: the rudder deflection (δ_{rc}), differential brake pressure (δP_{brkc}) – difference between right $P_{brk}MGR_c$ and left $P_{brk}MGL$ main gear pressures, and nose wheel deflection (θ_{NWC}). The actual control signal $u(t)$ is related to the virtual control $v(t)$ by equation (2), where B is the control effectiveness.

$$Bu(t) = v(t) \quad (2)$$

The solution of the problem (2) is found using the least-squares method (3)

$$u(t) = WB^T(BWB^T)^{-1}v \quad (3)$$

Taking into account the effective actuator limits $u^p_{lim_i}$ (saturation limits), the control weights W are chosen such that the control power is minimized (first term) (4), while the actuators tend to reach saturation at the same time (second term):

$$w_{i\in\{1,\dots m\}}(t) = \left(u^p_{lim_i}\right)^2 + \left(\frac{1}{b_i} - u^p_{lim_i}\right)|u_i - T| \quad (4)$$

The control allocator has two modes of operation:

- Restricted mode – the rudder and nose wheel are primary actuators while the brakes are secondary, only used when the primary actuators saturate (to minimize ground holding time).
- Unrestricted mode – all actuators are used at all times [4].

3. SIMULATION SETUP

3.1 FMI Integration and Co-Simulation Configuration

In the FMI configuration, the Simulink controller FMU is imported into Amesim using the FMU import assistant. The I/O signals are mapped to corresponding model variables:

Table 1. FMI block INPUTS

Variable	**Description**	**Unit**
RunwayVelocityX	Longitudinal velocity along runway	m/s
RunwayVelocityY	Lateral velocity along runway	m/s
YawAngle (ψ)	Aircraft yaw angle relative to runway	deg
YawRate (r)	Angular velocity around vertical axis	deg/s
RunwayDistanceY	Lateral position relative to runway centerline	m
AircraftVelocityX (body)	Forward velocity (body axis)	m/s
AircraftVelocityY (body)	Lateral velocity (body axis)	m/s

Table 2. FMI block OUTPUTS

Variable	**Description**	**Unit**
RudderDeflectionCmd	Rudder deflection command	den
NoseWheelAngleCmd	Nose-wheel steering command	deg
LeftBrakePressureCmd	Left main-gear brake pressure	bar
RightBrakePressureCmd	Right main-gear brake pressure	bar
TotalBrakePressureCmd	Average braking command	bar

The H-Infinity controller receives inputs comprising measured aircraft states, including longitudinal and lateral velocities, yaw angle and rate, and lateral position offset, which represent the instantaneous ground state and are continuously updated during co-simulation.

Based on these signals, the controller computes actuator commands for rudder deflection, nose-wheel angle, and brake pressure distribution.

These control signals are transmitted back to the Amesim model via the FMI interface at each time step, directly influencing the mechanical subsystems.

Each signal is mapped to its corresponding actuator input node, ensuring accurate integration of the control signals into the aircraft's dynamic model and affecting yaw rate derivative (angular acceleration) / yaw moment.

In the reverse Co-Simulation setup, an Amesim interface block is generated via S-function and connected to MATLAB/Simulink, enabling execution control from MATLAB. In both configurations, solver synchronization ensures identical integration steps and data exchange intervals.

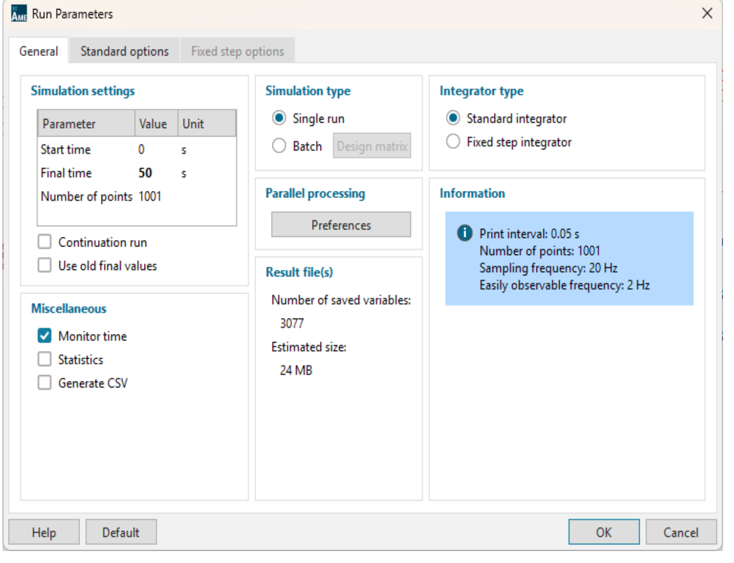

Fig. 5 Run parameters of the Amesim simulation

The co-simulation was executed in closed-loop mode using Amesim as the master solver and MATLAB/Simulink as the FMU slave. A fixed integration step of 0.001 s was selected to ensure synchronized data exchange between platforms.

The total simulation time was set to 50 s, sufficient to capture the full braking and stabilization maneuver.

All parameters, including solver settings and communication intervals, were aligned to maintain numerical consistency and stable convergence throughout the run. [5]

3.2 Simulation Scenario

The simulation represents a straight-line braking and stabilization maneuver on a dry, level runway. The aircraft starts with an initial longitudinal velocity of 61.7 m/s (120 kt) and an altitude of 1.7 m above the ground reference plane, restricted mode (the rudder and nose wheel are primary actuators while the brakes are secondary, only used when the primary actuators saturate) and no wind.

All angular rates and attitude angles are initialized at zero, representing a trimmed and balanced ground roll condition at the beginning of the simulation.

A small longitudinal offset of 5×10^{-5} degrees is applied to the initial position, corresponding to approximately 5.5 meters of displacement at the simulation latitude, effectively introducing a lateral deviation from the runway centerline. This offset serves as a controlled disturbance, allowing the controller to demonstrate its ability to realign the aircraft to the centerline during ground roll.

4. RESULTS AND DISCUSSIONS

The integrated H-Infinity control system demonstrated robust performance under both nominal and disturbed conditions.

The aircraft recovered from a 5.5 lateral deviation within 10 seconds, maintaining smooth actuator transitions.

The control allocator effectively distributed efforts, prioritizing rudder and nose wheel deflection, with differential braking engaged only under large yaw rate errors.

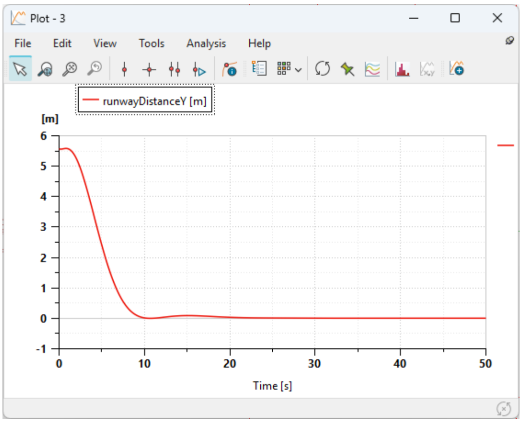

Fig. 6 Runway distance on Y axis [m]

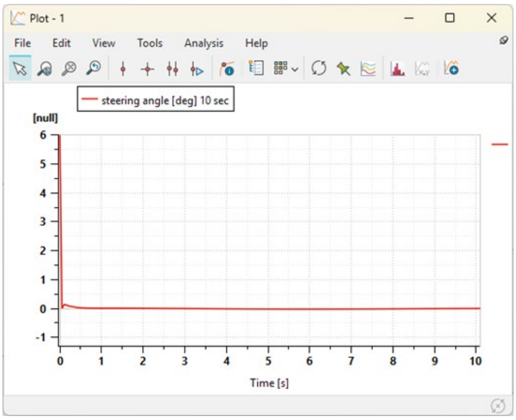

Fig. 7 Steering angle [deg]

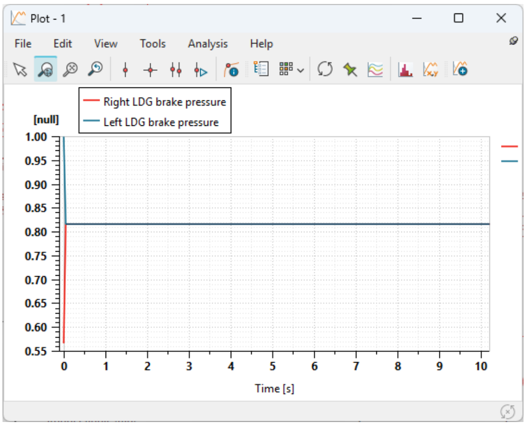

Fig. 8 Right and left landing gear braking pressure

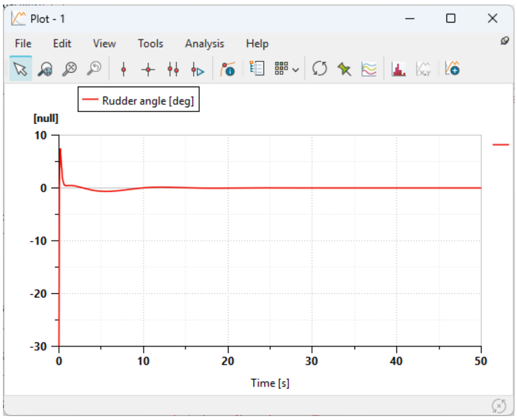

Fig. 9 Rudder angle command [deg]

This tracking performance is achieved through the effective coordination of the control allocator, which prioritizes rudder and nose wheel deflection for primary control authority. The actuator commands (Fig. 7 and Fig. 8) confirm smooth transitions, with differential braking engaged only during the initial phase to counteract the significant yaw rate error induced by the deviation.

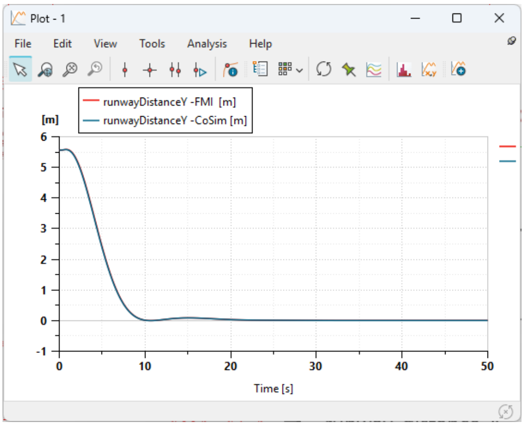

Fig. 10 Runway distance on Y axis FMI and CoSim comparison [m]

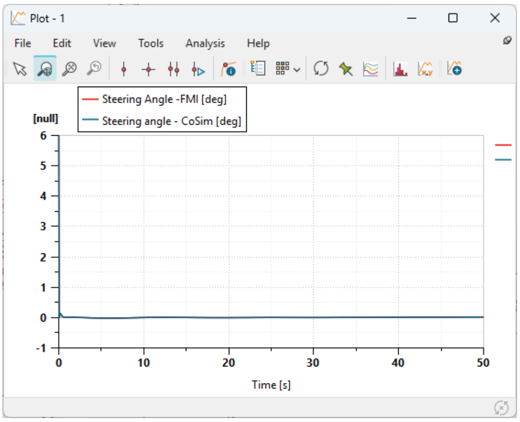

Fig. 11 Steering angle FMI and CoSim comparison [deg]

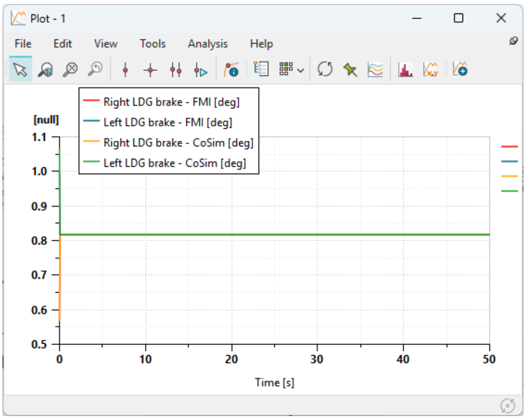

Fig. 12 LDG differential braking FMI and CoSim comparison

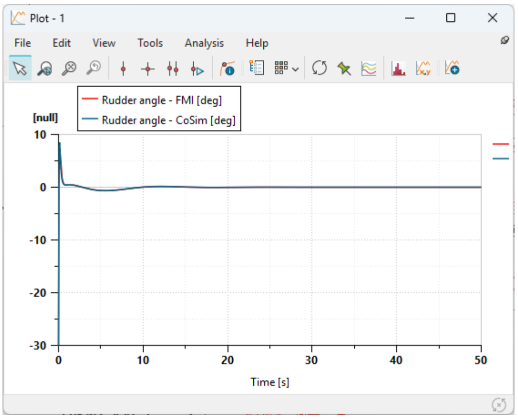

Fig. 13 Rudder angle command FMI and CoSim comparison [deg]

Fig. 10 to Fig. 13 compare the results obtained using the FMI and Co-Simulation (CoSim) configurations for the aircraft ground roll control framework. The two setups show agreement across all evaluated parameters, confirming their numerical equivalence. Fig. 10 illustrates the lateral position evolution, where both simulations demonstrate a smooth recovery to the runway centerline within ten seconds after the initial offset.

The steering angle (Fig. 11) and rudder command (Fig. 13) exhibit almost identical trajectories, showing that actuator coordination and control signal transfer are accurately maintained between platforms. Similarly, the differential braking pressures (Fig. 12) display consistent torque distribution on both sides, validating the proper functioning of the control allocator.

Overall, the results confirm stable and coherent data exchange between Amesim and Simulink, with no observable discrepancies in dynamic behaviour, demonstrating the robustness and reliability of the hybrid FMI–CoSim implementation.

5. CONCLUSIONS

The developed co-simulation framework successfully integrates a detailed Amesim ground dynamics model with a structured H-Infinity controller from Simulink through the FMI standard. The comparison between FMI and CoSim configurations confirms consistent numerical performance and seamless interoperability. The controller achieved effective runway centerline tracking and lateral stability by coordinating the rudder, nose-wheel, and differential braking actuators. Minimal differences between the two configurations validate accurate data synchronization and solver coupling. The framework proves to be flexible, reliable, and suitable for advanced analysis of aircraft ground dynamics. Furthermore, the FMI-based approach enhances model reusability and enables cross-platform validation of control strategies. These results establish a solid foundation for future research into adaptive and intelligent ground control systems.

ACKNOWLEDGMENTS

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI - UEFISCDI, project number PN-IV-P8-8.1-PME-2024-0030, within PNCDI IV.

REFERENCES

[1] * * * *Siemens Industries Digital Software. Simcenter Amesim*, version 2504, Siemens 2025

[2] * * * *MATLAB version:R2024a*, Natick, Massachusetts: The MathWorks Inc.; 2024

[3] T. Blochwitz, M. Otter, J. Åkesson, M. Arnold, C. Clauss, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson & A. Viel, Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In *Proceedings of the 9th International Modelica Conference* (pp. 173-184), 2012.

[4] E. Sadien, C. Roos, A. Birouche, M. Carton, C. Grimault, L. E. Romana, M. Basset, A simple and efficient control allocation scheme for on-ground aircraft runway centerline tracking, *Control Engineering Practice*, Volume **95**, 2020

[5] J. Li, & Z. Zhao, & C. Zhou, & X. Li, & H. Li, Pneumatic ABS Modeling and Failure Mode Analysis of Electromagnetic and Control Valves for Commercial Vehicles, *Electronics*, **9**. 318. 10.3390/electronics9020318, 2020.