2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), 29 September - 3 October 2024, San Diego, CA, USA
DOI:10.1109/DASC62030.2024.10749078

A Certifiable AI-Based Braking Control Framework for Landing Using Scientific Machine Learning

Mevlut Uzun, Ugurcan Celik, Guney Guner, Orhan Ozdemir, Gokhan Inalhan
Autonomy and AI, Cranfield University
Cranfield, United Kingdom
Email : {mevlut.uzun, ugurcan.celik, guney.guner, orhan.ozdemir, inalhan}@cranfield.ac.uk

***Abstract*—This paper proposes an AI-based braking control system for aircraft during landing. Utilizing scientific machine learning, we train an agent to apply the most effective braking strategy under various landing conditions. This approach ensures physically consistent outputs by grounding the algorithm in the principles of landing physics. Our results demonstrate that the aircraft can successfully decelerate without skidding across all runway conditions and landing speeds. Additionally, the algorithm maintains performance and safety even when brake performance degradation and initial yaw angles are introduced. This robustness is crucial for the certification of AI in safety-critical systems, as the proposed framework provides a reliable and effective solution.**

***Index Terms*—aircraft braking, scientific machine learning, certification, safety critical systems**

I. Introduction

The current braking control system, reliant on rule-based logic, is nearing its operational limits due to the escalating complexity of aircraft systems over the last two decades. Furthermore, with the advent of next-generation aircraft, adhering to the traditional route of rule-based logic for executing such critical functions will become infeasible. Therefore, there is a pressing need to integrate an AI framework into the aircraft to undertake this task, which could be trained off-board in a safe manner.

According to the statistical analysis of commercial aviation accidents by Airbus [1], over the last two decades, 12% of all fatal accidents and 59% of non-fatal hull losses occurred during flight phases that rely heavily on landing gear and braking control performance. Reliable and resilient performance of the landing gear system in challenging environmental conditions, as well as in the face of system/component failures or malfunctions, is crucial for overall safety and the commercial feasibility of operations. However, current landing gear designs and braking control systems are limited to classical control strategies with little to no parametric adaptation and reconfiguration capability, following a "one solution fits all" approach. It is also important to note that current designs are further disadvantaged by their minimal access to and usage of available aircraft sensing and health monitoring data.

Aircraft braking control is a topic that has been discussed previously, yet there are several studies in the literature. The previous works have usually focused on designing an anti-skid braking system. The first approaches were usually rule-based algorithms [2], and then the trend shifted to designing conventional controllers. Jiao et al. [3] designed a traditional controller for controlling the brake pressure, supported by runway identification and a neural network-based aerodynamic model. The same authors later proposed a heuristic controller to maximize the friction coefficient [4]. Another work done by D'Avico et al. observed aircraft dynamics through experimental flight data and proposed a state machine-based control framework [5], [6]. Another study by Li [7] applied a standard PID controller for anti-skid. With the advancement of machine learning, some researchers used reinforcement learning to this problem. Liu et al. [8] proposed a twin delayed deep deterministic policy gradient algorithm, considering the aircraft's final speed, simulation time, and slip ratio as reward functions. In a similar method, Radac [9] used standard Q-learning to tackle the problem. The braking control problem is not unique to aerospace, as it has been widely studied for ground vehicles, applying similar control strategies [10], [11], [12], [13]. Even though these studies reveal promising results, they do not consider either brake performance degradation or other failure cases. Additionally, they are tuned for specific conditions, making them inefficient for all flight conditions. In this paper, we tackle the problem with a recently emerging trend named scientific machine learning, which leverages powerful sides of both model-based approach and neural networks [14], [15].

The certification of AI systems in safety-critical aerospace applications is a complex process that requires safety, reliability, and performance standards to be met. Regulatory bodies and industry groups have developed comprehensive frameworks and guidelines to address the unique challenges posed by AI in aviation. Key frameworks include the collaboration between EUROCAE WG-114 and SAE G-34, which produced the SAE AIR6988 document. This document outlines the current gaps in aerospace standards and proposes the necessary steps to develop a certification standard for AI systems in aviation (SAE International). Additionally, the European Union Aviation Safety Agency (EASA) has published an AI Roadmap that provides guidance on integrating and certifying AI technologies in aviation, ensuring that these systems meet stringent safety and operational requirements through rigorous testing and validation processes. The European Union Aviation Safety Agency (EASA) has provided comprehensive guidance for the certification of Level 1 and Level 2 machine learning (ML) applications, which is essential for integrating AI into aviation systems [16] [17]. The EASA AI Roadmap is

divided into phases, with the current guidance focusing on Level 1 AI applications (assistance to human) and Level 2 AI applications (human-AI teaming). The roadmap aims to establish a common policy applicable across the EU civil aviation framework, ensuring that AI applications meet high safety standards and trustworthiness requirements. Another work by EASA and Collins Aerospace proposes methodology for learning assurance [18]. As certification of AI in aerospace systems is still an ongoing task, there is not many studies with use cases yet. A good example is work done by Coffer [19], where the authors proposes a verification methodology for a collision avoidance algorithm trained by reinforcement learning.

In this paper, after presenting the results, we will reveal how the developed algorithm responds to the objectives defined by EASA and provides AI assurance.

This paper has two main contributions: First, we present an AI framework based on scientific machine learning that does not require any labeled data and adapts the braking policy to various flight conditions and braking failures. Second, we map the algorithm's features to the certification objectives defined by EASA to demonstrate its potential for integration into a safety-critical system.

The rest of the paper is organized as follows: In Section II, we explain how the braking system works through aircraft and wheel dynamics during landing. Section III is dedicated to explaining the scientific machine learning framework and its integration into the braking problem. Section IV presents the results for various flight and failure conditions. In Section V, we discuss how the AI-based algorithm meets the essential certification objectives. Finally, Section VI concludes the paper.

II. Problem Definition and Objective

A. Aircraft Braking System Overview

Aircraft brakes work primarily to slow down and stop the aircraft during landing, as well as to control the aircraft's speed and maintain its position during taxiing and parking on the ground. They are essential for ensuring the safety of the passengers, crew, and aircraft. The braking system on most modern aircraft consists of several components: disc brakes, hydraulic system, anti-skid system, autobrakes, and other decelerating elements such as reverse thrust and spoilers.

Most aircraft use disc brakes, similar to car brakes but more robust for aviation demands. Disc brakes consist of rotating discs (attached to the wheel), brake pads, and calipers. When applied, the calipers press the pads against the discs, creating friction to slow the wheel and aircraft. Modern designs use hydraulic disk-based systems with either single or multiple disk configurations. The size and number of disks depend on the required torque, while the disk's thickness and mass relate to the energy retention within the brake. The heat generated at the pad-disk contact point is primarily stored in the disk

Hydraulic fluid transfers force from the brake pedals in the cockpit to the brake calipers on the wheels. Pressing the brake pedals generates pressure in the hydraulic system, causing the calipers to clamp the brake pads onto the discs (Figure 1). Most aircraft have an anti-skid system to prevent wheel lock-up, which could lead to loss of control or tire damage. This system monitors wheel speed and modulates brake pressure to maintain control during heavy braking. Some commercial aircraft feature autobrakes, allowing the pilot to pre-select a braking force level for landing. Upon touchdown, the autobrake system automatically applies the selected braking force, useful for high-speed or slippery landings. Additionally, reverse thrust can be used alongside brakes to slow the aircraft more effectively by directing engine thrust forward, aiding deceleration during landing and on slippery runways. More detailed landing gear and braking system analysis can be found in [20] [21] [22] [23].

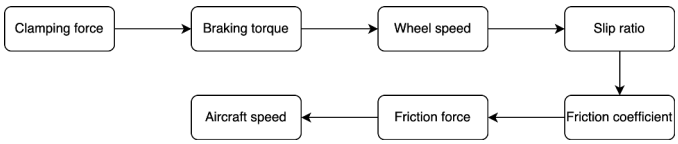

Figure 1. Braking actuation process

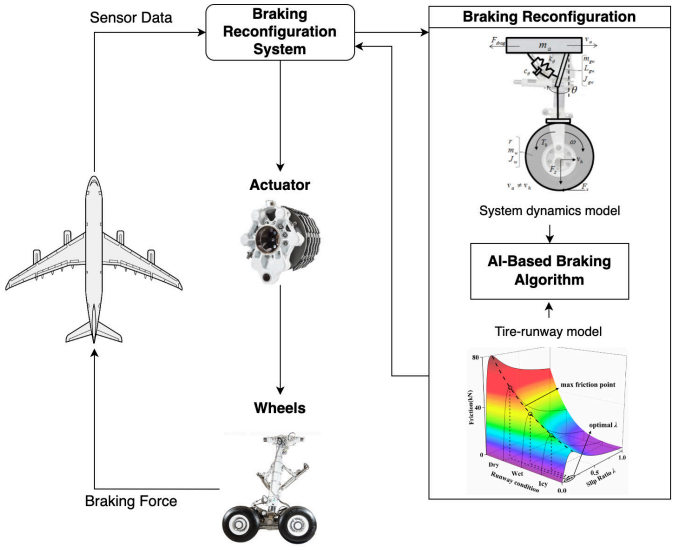

Figure 2. AI-based braking overview (The small images used in the diagram are sourced from various references [3] [24] [5])

Figure 2 depicts the integration of AI-based methods into the aircraft braking reconfiguration system. The figure illustrates an aircraft along with the sensor data utilized by the reconfiguration system. The system takes into account various dynamic models, including those of the aircraft, wheels, brakes, and tire-surface friction. By leveraging reinforcement learning algorithms, the system intelligently selects the optimal braking strategy based on the available sensor data. The chosen command is then transmitted to the brake actuators, which subsequently apply the appropriate braking torque to the wheels. This coordinated action of the actuators effectively slows down the aircraft, resulting in a closed-loop system that

continuously adjusts and optimizes the braking process.

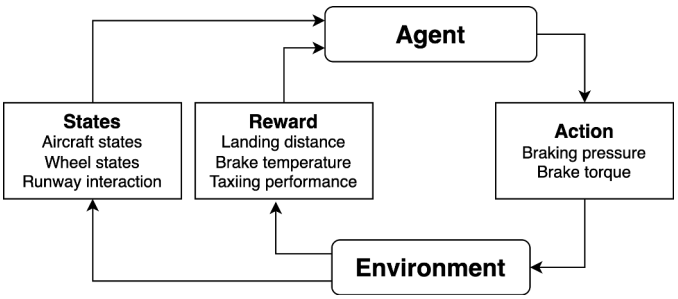

Figure 3. AI-based braking states, actions and rewards

B. Aircraft Braking Model

According to Figure 4, it is apparent that the primary longitudinal forces acting on the aircraft are the aircraft thrust T_0, the aerodynamic drag force D, and the friction generated between the two main wheels and the runway, F_{N1} and F_{N2}. Evidently, the movement of the aircraft in the longitudinal direction can be expressed using the following description:

$$\begin{aligned} m\dot{V}_{GS} &= T_0 - D - F_{f1} - F_{f2} - \alpha_R mg \\ L + F_{N1} + F_{N2} - mg &= 0 \end{aligned} \tag{1}$$

Where m is the aircraft mass, α_R is the runway slope. The friction forces are:

$$\begin{aligned} F_{f1} &= \mu_1 F_{N1} \\ F_{f2} &= \mu_2 F_{N2} \end{aligned} \tag{2}$$

where μ_1, μ_2 represent the tire-runway friction coefficients and F_{N1},F_{N2} are the normal forces. Normal forces increase during landing due to decreasing lift as the aircraft loses speed. This is also reflected in the dynamics:

$$\begin{aligned} F_N &= W - L \\ L &= \frac{1}{2} C_D \rho V_{TAS}^2 S \end{aligned} \tag{3}$$

where W is the aircraft weight, C_L is the lift coefficient at landing flap configuration.

There are several modelling approaches for the thrust T_0. In general, it is a function of the aircraft speed: $T_0 = f(V)$. In the simplest form, $T_0 = T_0' + K_v V$ is the initial residual thrust, and K_v is the velocity coefficient. Drag force D can be modeled as:

$$D = \frac{1}{2} C_D \rho V_{TAS}^2 S \tag{4}$$

Where S is the wing reference area, V is the true airspeed (it is equal to the ground speed in landing), and C_D is the drag coefficient:

$$C_D = C_{D0} + C_{D2} C_L^2 \tag{5}$$

Where C_{D0} and C_{D2} are aircraft-specific drag coefficients, and C_L is the lift coefficient:

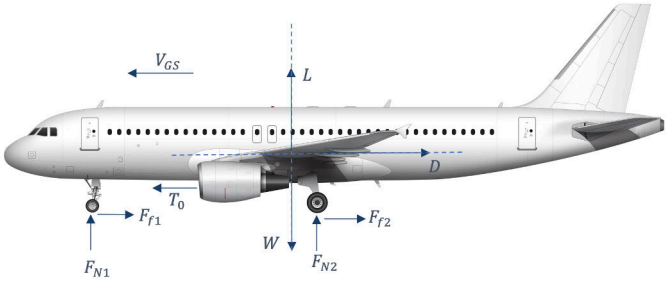

Figure 4. Forces acting on an aircraft during landing roll

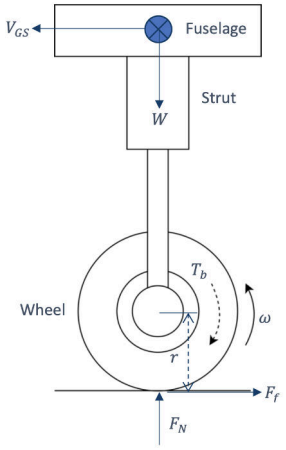

Figure 5. Forces and moments acting on aircraft wheels

$$C_L = \frac{2mg}{\rho V_{TAS}^2 S} \tag{6}$$

For the rest of the paper, aircraft speed will be denoted by V_{GS} and ground speed will be considered as equal to true airspeed V_{TAS}, as sea level conditions are assumed.

As shown in Figure 5, the diagram represents the various forces acting on the wheel, including the weight of the aircraft, the normal force exerted by the ground, and the frictional force between the wheel and the ground. Using the force balance diagram, we can derive the dynamic equation of the wheel, which describes how the wheel responds to these forces and how it behaves in motion. The dynamic equation considers various factors, such as the moment of inertia of the wheel, the angular velocity of the wheel, and the torque acting on the wheel.

$$J_w \dot{\omega} = (F_t - F_f) \cdot r_w - T_b = F_t \cdot r_w - \mu \cdot F_N \cdot r_w - k_b P_b \tag{7}$$

The variables involved in the dynamic equation of the aircraft's main wheel are as follows: J represents the wheel's moment of inertia, r_w denotes the wheel's radius, w is the wheel's angular velocity, F_t is the traction force, and T_b represents the braking torque. Additionally, P_b is the brake pressure, while k_b is either the brake disc torque-pressure

coefficient or the friction coefficient. The value of k_b is defined as the ratio of the braking torque T_b to the brake pressure P_b i.e., $k_b = T_b/P_b$.

Assuming all engines produce the same thrust, drag is equally distributed along the wings, and no aileron or rudder displacement, the only remaining factor affecting the aircraft yaw on the runway is the friction force:

$$\dot{\psi} = \frac{(F_{f1} - F_{f2})\, l_w}{I_{zz}} \tag{8}$$

where l_w is the lateral distance from the center-line of the aircraft to the center of gravity, I_{zz} is the moment of inertia about the yaw axis.

The amount of torque produced by a brake is not solely dependent on the clamping pressure applied, but also changes with variations in wheel speed and brake temperature. Thus, it is essential to guarantee that the brake can deliver a minimum amount of torque to achieve the desired performance of the aircraft. Several factors impact the amount of torque that a brake can generate, including the coefficient of friction at the contact surfaces, the applied clamping pressure, and the mechanical losses that occur due to friction while clamping the rotors and stators. To calculate the brake's produced torque, the following formula is used:

$$T_b = 2N_R \cdot \mu_b \times F_b \times r_b \tag{9}$$

where mu_b is the brake efficiency, which includes both the coefficient of friction of the brake material and the mechanical losses within the brake, N_R is the number of rotors, F_b is the clamping force, and r_b is the brake disk radius.

Understanding and accurately modeling the interaction between the aircraft tires and the runway surface is essential for predicting braking distances, optimizing braking efficiency, and ensuring the aircraft's stability during landing and ground operations. A key parameter in this modeling process is the friction coefficient between the tire and the runway surface, which directly affects the braking capacity and the tire's ability to grip the runway. To quantify the relative slip motion between the wheels and the runway, a slip ratio λ is defined:

$$\lambda = (V_{GS} - \omega r_w)\,/V_{GS} \tag{10}$$

The coefficient representing the interaction between the tire and the runway is influenced by several factors, such as current runway conditions, aircraft speed, and slip rate. Figure 6 illustrates the friction coefficient evolution with slip ratio for various runway conditions. To calculate this coefficient, a commonly used empirical formula developed by Pacejka [25] is widely applied. The formula can be expressed as follows:

$$\mu\,(\lambda, \tau_j) = \tau_1 \sin\left(\tau_2 \operatorname{arctg}\left(\tau_3 \lambda\right)\right) \tag{11}$$

where $\tau_j\ (j = 1, 2, 3)$ represents the peak factor, stiffness factor, and curve shape factor, respectively. The function $\mu\,(\lambda, \tau_j)$ calculates the coefficient based on the slip ratio λ and the specified parameters τ_j.

Runway condition	τ_1	τ_2	τ_3
Dry	0.85	1.5344	14.5
Wet	0.40	2.0	8.2
Snow	0.28	2.0875	10

Table I
RUNWAY CONDITIONS AND CORRESPONDING τ VALUES

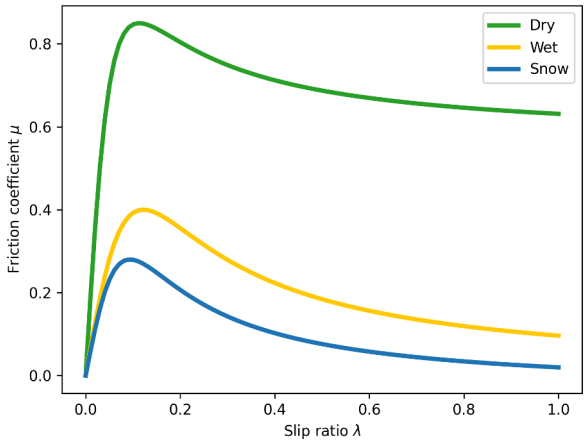

Figure 6. Friction coefficient at various runway conditions

III. METHODOLOGY

This section presents the framework for developing AI-based braking control for a landing aircraft. The baseline braking policy is formulated using scientific machine learning.

A. Problem Formulation

In this work, our braking model does not incorporate the full dynamics of the hydraulic system. It considers aircraft speed V, wheel speed w and aircraft yaw ψ as the states, where our inputs are brake pedal positions. We assume two brakes on the aircraft, representing two landing wheel groups on left and right side of the aircraft. The state vector $\mathbf{x}$ and the input vector $\mathbf{u}$ are defined as:

$$\mathbf{x} = \begin{bmatrix} V \\ w_1 \\ w_2 \\ \psi \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

The system dynamics are expressed through $\dot{x} = f\,(x, u)$, and the braking control policy is defined by π:

$$\pi(\mathcal{X}(t), W_{NN}) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \approx \mathcal{NN}(\mathcal{X}(t), W_{NN}) \tag{12}$$

where $\mathcal{NN}$ is the neural network to be trained, $\mathcal{X}$ is the input vector to the neural network, and W_{NN} represents the neural network's trainable parameters. The objective is to find the optimum braking policy π^*, which minimizes a loss function defined as $\mathcal{L}$:

Nominal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Touchdown Speed (m/s)	60	60	60	70	70	70
Runway Condition	Dry	Wet	Snow	Dry	Wet	Snow
Landing distance (m)	590	757	966	823	1128	1370

Table II
LANDING SCENARIOS WITH VARIOUS RUNWAY CONDITIONS, NOMINAL BRAKE PERFORMANCE

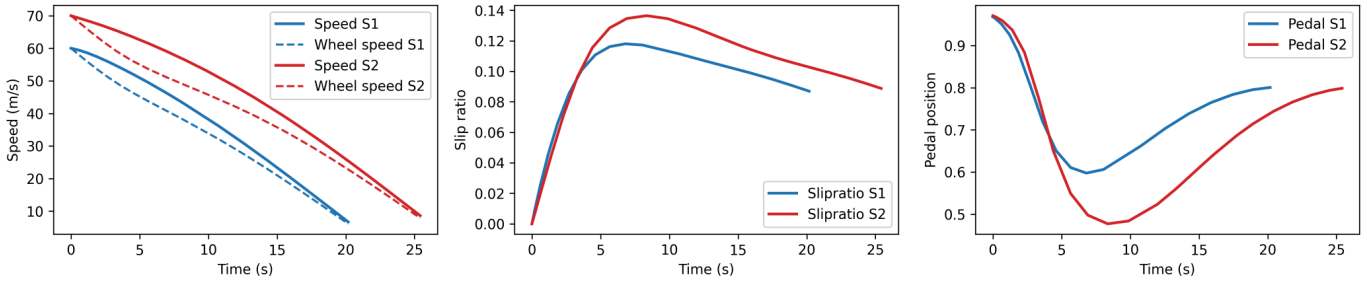

Figure 7. Wet runway, both brakes at nominal conditions

$$\pi^* = \arg\min_{\pi} \mathcal{L}(x(t), \dot{x}(t), u(t)) \tag{13}$$

where the loss function's $\mathcal{L}$ details will be explained in the next sections.

B. Scientific Machine Learning

The principle of integrating governing physics into the training processes of machine learning models is to enhance their generalization capabilities. By this approach, the concept of scientific machine learning, implemented using Universal Ordinary Differential Equations (UODEs), has been proposed and formalized in [26] for scientific model discovery. Figure 8 depicts the general training framework for scientific machine learning:

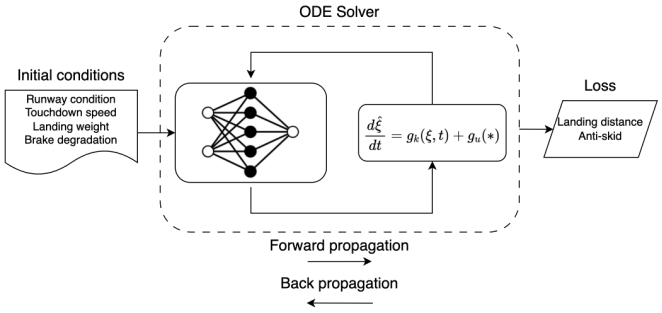

Figure 8. Scientific machine learning flow for AI-based braking control

The principle of Universal Ordinary Differential Equations (UODEs) is to combine a mechanistic model with a universal approximator to estimate unknown dynamics. A basic first-order ODE, given by the ground truth model $\frac{d\xi}{dt} = f(\xi, t)$ can be transformed into a UODE form as $\frac{d\hat{\xi}}{dt} = g_k(\xi, t) + g_u(*)$ Here, $g_k(\xi, t)$ denotes the known part of the system, while $g_u(*)$ represents the unknown dynamics to be discovered. This unknown component is approximated using a neural network, $g_u(*) \approx \mathcal{NN}(\mathcal{X}, W_{NN})$ During forward propagation, the ODE system is simulated by solving it, and a scalar loss function, such as the mean squared error between the ODE solution and experimental data, is calculated. In backward propagation, continuous-time gradients of the network parameters with respect to the loss function are computed using adjoint sensitivity methods [26], [27], [28]. After computing the gradients, network parameters are updated using gradient descent methods.

Algorithm 1 Scientific machine learning minibatch training algorithm

Input: ODE Solver(ODE, IC's), Epoch Size: N, Mini-Batch Size: M
Data: Initial Landing Conditions $\{V_{GS}, r, \psi\}$
Initialization: Initilization of Network Weights, θ_{NN}
Optimization Variable:, W_{NN}
for $epoch = 1, 2, \ldots, N$ **do**
  Generate Minibatches
  for $Minibatch = 1, 2, \ldots$ **do**
    for Landing IC's in Minibatch **do**
      $\eta_1, \eta_2 \leftarrow$ Brake degradation
      $x(t)_i, u(t)_i \leftarrow$ Run the $simulation(IC_i, x^E(t_0), W_{NN})$
      Compute Loss, $\mathcal{L}_i \leftarrow x(t)_i$
      Compute Gradient, $\left(\frac{d\mathcal{L}_i}{dW_{NN}}\right)_i \leftarrow \mathcal{L}_i, x(t)_i, u(t)_i, W_{NN}$
    end for
    Compute Mean Loss, $\mathcal{L}_{mean} = \frac{1}{M}\Sigma_{i=1}^{M}\mathcal{L}_i$
    $Compute\ Mean\ Gradient$:
      $\left(\frac{d\mathcal{L}}{dW_{NN}}\right)_{mean} = \frac{1}{M}\Sigma_{i=1}^{M}\left(\frac{d\mathcal{L}_i}{dW_{NN}}\right)_i$
    $Gradient\ Descent\ with\ Adam\ Optimizer$:
      $W_{updated} \leftarrow \mathcal{L}_{mean}, \left(\frac{d\mathcal{L}}{dW_{NN}}\right)_{mean}$
    Update Network Weights, $W_{NN} \leftarrow W_{updated}$
  end for
end for

1 explains the flow of the training algorithm. We first start by assigning a matrix of initial conditions. The matrix comprises of all possible initial condition tuples given the available values of each state. We take touchdown ground speed $V_{GS} \in [60, 70][m/s]$, runway condition $r \in \{dry, wet, snow\}$, aircraft yaw $\psi \in [-10, 10][deg]$. Then the simulation is run through the ODE solver and using the current neural network

$\mathcal{NN}$ as the braking control policy. After the simulation is terminated, loss function $\mathcal{L}$ is calculated by providing the solution $x(t)$. Through adjoint sensitivity methods, backpropagation and gradient calculation are completed. Finally, the network parameters W_{NN} are updated. Simulation termination conditions are defined as:

$$\begin{cases} V_{\text{GS}} \leq 10\,\text{m/s} \\ t \geq 60\,\text{s} \end{cases}$$

At each minibatch, we sample random values for brake performance degradation factor to cover faulty scenarios:

$$\begin{aligned} \eta_1 &= 0.5 + X_1 \\ \eta_2 &= 0.5 + X_2 \\ X_1, X_2 &\sim \mathcal{U}\,(0, 0.5) \end{aligned} \tag{14}$$

For simplicity during neural network training, we consider the inputs to be pedal positions, taking range between 0 and 1, $u_{1,2} \in [0, 1]$. Then we convert these values into braking forces by multiplying them with a constant, $F_{b,1} = u_1\kappa$, $F_{b,2} = u_2\kappa$. As defined in equations above, these then converted into braking torque, and transferred into wheel speed dynamics.

IV. Results

In this section, we present several scenarios where the aircraft decelerates from various touchdown speeds, under different runway conditions, and with varying levels of brake performance degradation. The goal is to demonstrate that the algorithm performs well under all these conditions. This is crucial for AI assurance during the certification process, as it shows that the algorithm consistently yields reliable outputs. By simulating brake performance degradation, we illustrate that even when brakes do not respond as they do under nominal conditions, the algorithm can still adapt and calculate appropriate brake inputs to ensure smooth deceleration over the shortest possible distance, without compromising safety.

A. Case 1: Nominal landing conditions

In the first scenario, the aircraft lands at either 70 m/s or 60 m/s. All brakes operate under nominal conditions, meaning there are no issues with the hydraulic system, and both brakes can produce their maximum braking torque. The algorithm is then tested under all runway conditions. In this scenario, the loss function defined as minimizing distance while avoiding skid conditions:

$$\mathcal{L} = \cdot \sum_i \lambda_1 d_i + \lambda_2 \left(\|ReLU\left(-w_{i,1}\right)\|^2 + \|ReLU\left(-w_{i,2}\right)\|^2 \right) \tag{15}$$

where i represents each minibatch, i.e. each initial condition for this problem. d_i is the landing distance and w_i are the wheel speeds for the corresponding initial condition, respectively. We penalize negative wheel speeds as they physically indicate that aircraft is skidding. $\lambda_{1,2}$ are the hyper-parameters of the loss components. II reveals the landing distances for each touchdown and runway condition pair. As expected, the shortest landing distances are on dry runways, and they tend to increase with more contaminated runways. Figure X represents the aircraft speeds, wheel speeds, slip ratios and brake control inputs. $S1$ and $S2$ correspond to initial speeds of $60[m/s]$ and $70[m/s]$, respectively.

Figure 7 depicts an example for wet runway conditions, where both brakes operate without degradation. As shown in the figure, the slip ratios never exceed the critical skidding condition, which is usually considered to be 0.7. Moreover, the algorithm tries to maintain a slip ratio around 0.1, which approximately maximizes the friction coefficient μ, as illustrated in Figure 6. Table II reveals the landing distances, which tend to increase with either increasing touchdown speed or more contaminated runways.

B. Case 2: Brake performance degradation

In the third scenario, the aircraft lands at either 70 m/s or 60 m/s. Both brakes have the same level of performance degradation, meaning their maximum producible torque is reduced to 80%. The algorithm is then tested under all runway conditions. We apply degradation as a multiplier to the braking force F_b:

$$\begin{aligned} F_{b,1} &= \eta_1 u_1 \kappa \\ F_{b,2} &= \eta_2 u_2 \kappa \end{aligned} \tag{16}$$

where the degradation factor is defined as $\eta \in [0, 1]$, meaning that a value of 1 represents fully functional brakes, while a value of 0 means no torque produced by the brake. Figures 9 , 10, and 11 reveal speed profiles, slip ratios, and brake pedal inputs for dry, wet, and snow conditions with a brake performance of 80% for both brakes. As can be seen in the figures, the algorithm successfully decelerates the aircraft in all conditions without skidding. Compared to nominal conditions, the algorithm increases the input magnitude along the landing to compensate for torque loss due to brake performance degradation. In all runway conditions, the algorithm maintains a slip ratio close to 0.1, which maximizes the friction coefficient.

C. Case 3: Asynchronous brake performance degradation

In the fourth scenario, the aircraft lands at either 70 m/s or 60 m/s. This time, the brakes have different levels of degradation. This includes the case where one brake is fully functional while the other is degraded. It also covers scenarios where both brakes are degraded, but to different extents. The aim is to demonstrate that even though the algorithm is unaware of the specific degradation levels, it can adapt to this asynchronous situation. The algorithm is then tested under all runway conditions.

In this case we add a third element to the loss function to represent the torque generated by the misbalance between two friction forces generated by two brakes:

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Touchdown	60	60	60	70	70	70
Runway	Dry	Wet	Snow	Dry	Wet	Snow
Landing	704	802	979	978	1186	1415

Table III
LANDING SCENARIOS WITH VARIOUS RUNWAY CONDITIONS, BRAKE 1 AT 80% PERFORMANCE, BRAKE 2 AT 80% PERFORMANCE

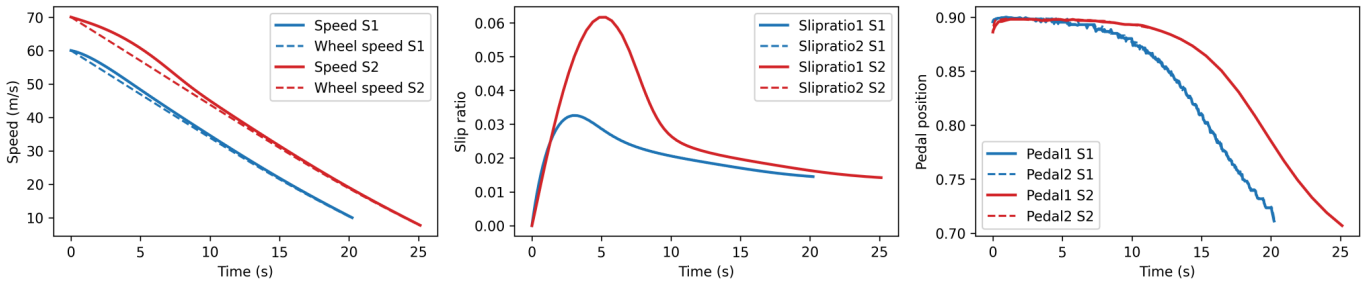

Figure 9. Dry runway, brake 1 at 80% performance, brake 2 at 80% performance

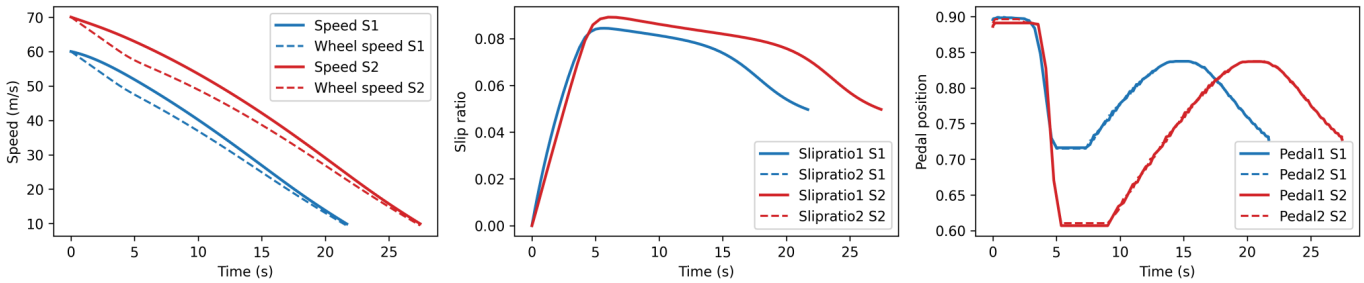

Figure 10. Wet runway, brake 1 at 80% performance, brake 2 at 80% performance

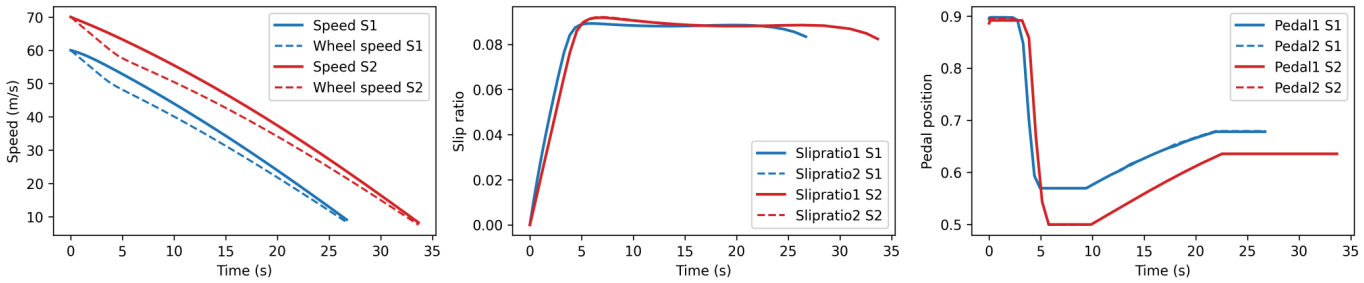

Figure 11. Snowy runway, brake 1 at 80% performance, brake 2 at 80% performance

$$\mathcal{L} = \sum_i \lambda_1 d_i + \lambda_2 \left(\| ReLU\left(-w_{i,1}\right) \|^2 + \| ReLU\left(-w_{i,2}\right) \|^2 \right) + \lambda_3 \sum |\psi_i| \quad (17)$$

Here with the additional term, we penalize any displacement in the yaw angle ψ, ensuring the aircraft does not spin by producing equal friction force from both brakes.

We have demonstrated several brake performance degradation cases. In the first scenario, we consider 90% and 80% for brakes 1 and 2, respectively. Second, we assume the first brake operates fully functional where the second brake operates at 80%. Finally, we consider even more extreme case where the first brake is fully operational again, but the second brake is at 50% performance.

Figure 12 shows the first scenario on a snowy runway. As in previous cases, the aircraft is successfully decelerated. The key difference from the equal or no degradation cases is that the brake positions are not identical. To prevent aircraft spin due to imbalance in braking forces, the algorithm applies different braking levels to each brake. Since the first brake is

89	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 5**	**Case 6**
Touchdown	60	60	60	70	70	70
Runway	Dry	Wet	Snow	Dry	Wet	Snow
Landing	673	789	980	959	1174	1409

Table IV
LANDING SCENARIOS WITH VARIOUS RUNWAY CONDITIONS, BRAKE 1 AT 80% PERFORMANCE, BRAKE 2 AT 90% PERFORMANCE

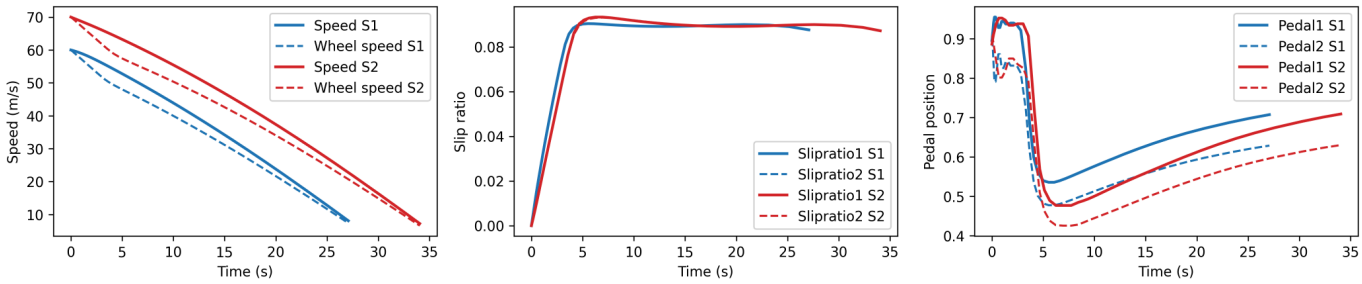

Figure 12. Snowy runway, brake 1 at 80% performance, brake 2 at 90% performance

more degraded, it receives more input to achieve the required braking force. On the other hand, the second brake, which is less degraded, receives less input. Thus, the total braking torque is limited by the performance of the most degraded brake. Table IV summarizes the landing distances for various runway conditions. The distances increase as the runway surface transitions from dry to snow. As expected, these distances are also longer than those under nominal braking conditions.

Figure 13 illustrates another example of snowy conditions, where the first brake operates fully, but the second brake operates at 80%. Similar to the previous case, the algorithm applies more input to the second brake to compensate for the torque loss, and less to the first brake to ensure no additional torque is generated on the aircraft's vertical axis. An extreme case is shown in Figure 14, where the second brake's performance is reduced to 50%. Despite this, the algorithm is able to decelerate the aircraft safely. Almost maximum input is applied to the second brake, while the input is around 0.5 for the first brake to balance the braking torques. Tables V and VI present the landing distances for a fully functional first brake, and with 80% and 50% degradation for the second brake. Among all scenarios, the longest landing distance and time occur on a snowy runway with 50% degradation in the second brake.

D. Case 4: Initial yaw at touchdown

In this final scenario, the first brake operates at 90% and the second at 80%. Additionally, we introduce an initial yaw, assuming a crosswind, during landing. As seen in Figure 15, even though there is an initial yaw angle of $5[deg]$ and brake performance mismatch, the algorithm decelerates the aircraft in both scenarios and aligns it with the runway.

V. CERTIFICATION OBJECTIVES AND AI ASSURANCE

In this section, we consider the certification objectives defined in the EASA AI Concept Paper [16]. Although we will not cover all the objectives in this paper, we will address the most important ones related to AI assurance. Certain objectives concerning the system's functional analysis and its mapping to standard operating procedures must be handled through collaboration with OEMs.

Objective CO-01:*The applicant should identify the list of end users that are intended to interact with the AI-based system, together with their roles, their responsibilities and their expected expertise (including assumptions made on the level of training, qualification and skills).*

The primary users of the AI-based braking system are the flight crew, typically comprising two members as in current commercial aircraft operations. Training and task distribution for these crew members are expected to remain similar to standard procedures. The crew member will monitor the system and have the ability to override it if necessary. Standard autobraking systems of current aircraft can be there as a backup.

Objective CO-02:*For each end user, the applicant should identify which high-level tasks are intended to be performed in interaction with the AI-based system.*

In the traditional two-person flight deck, tasks related to the AI-based system include activation, monitoring, and overriding its operations. The crew can deactivate the AI-based system if necessary.

Objective CL-01:*The applicant should classify the AI-based system, based on the levels presented in the document, with adequate justifications.*

For this version, we classify our AI-based braking algorithm as Level 2B - Human-AI cooperation, which corresponds to supervised automatic decision and action implementation. This means that users can actively monitor the tasks allocated to the AI system and intervene in its actions as necessary.

18	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Touchdown	60	60	60	70	70	70
Runway	Dry	Wet	Snow	Dry	Wet	Snow
Landing	658	781	977	916	1167	1412

Table V
LANDING SCENARIOS WITH VARIOUS RUNWAY CONDITIONS, BRAKE 1 AT NOMINAL PERFORMANCE, BRAKE 2 AT 80% PERFORMANCE

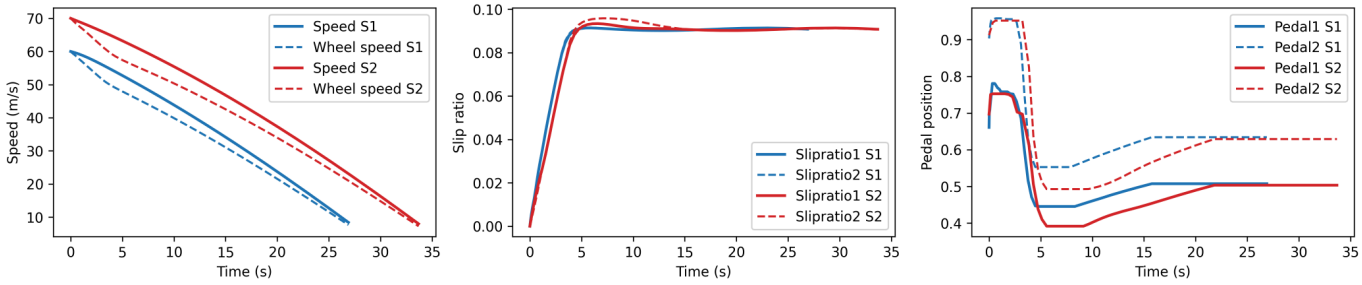

Figure 13. Snowy runway, brake 1 at nominal performance, brake 2 at 80% performance

15	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Touchdown	60	60	60	70	70	70
Runway	Dry	Wet	Snow	Dry	Wet	Snow
Landing	996	1031	1051	1366	1417	1471

Table VI
LANDING SCENARIOS WITH VARIOUS RUNWAY CONDITIONS, BRAKE 1 AT NOMINAL PERFORMANCE, BRAKE 2 AT 50% PERFORMANCE

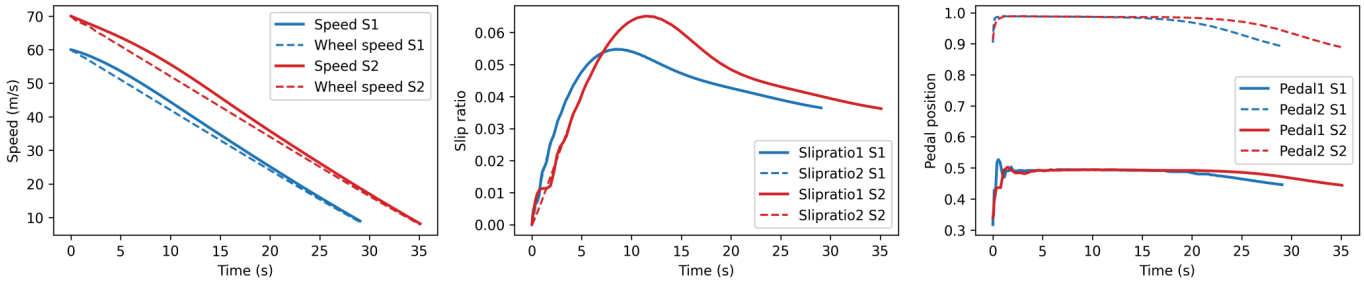

Figure 14. Wet runway, brake 1 at nominal performance, brake 2 at 50% performance

Objective SA-01:*The applicant should perform a safety (support) assessment for all AI-based (sub)systems, identifying and addressing specificities introduced by AI/ML usage.*

As illustrated in Section IV, the algorithm performs well under various failure conditions on dry, wet, and snowy runways. We have shown that even when there is a performance mismatch between the brakes, the algorithm is capable of identifying the situation and adapting accordingly. Additionally, if there is an initial yaw during touchdown due to crosswind, the algorithm can recognize it and apply the proper input to align the aircraft with the runway and complete the landing.

Objective SA-02:*The applicant should identify which data needs to be recorded for the purpose of supporting the continuous safety assessment.*

There is no need to install extra modules to observe the aircraft states necessary for monitoring this algorithm. The required parameters—aircraft speed, wheel speed, and yaw angle—are essential and are already being recorded by high-quality sensors.

Objective DA-03: *The applicant should define the set of parameters pertaining to the AI/ML constituent ODD, and trace them to the corresponding parameters pertaining to the OD when applicable.*

The input to the neural network consists of aircraft and wheel speed. As the algorithm is trained on the initial conditions gathered from historical flight data, it is unlikely to observe any out of distribution speed. If we still aim to cover whole feasible flight envelop during landing, these distributions can be easily introduced to the training process without amending anything.

VI. Conclusion

In this paper, we presented an AI-based braking control framework for landing aircraft. The algorithm offers efficiency by aiming to minimize landing distance and safety by acting as an anti-skid module and ensuring the aircraft remains aligned with the runway. We introduced a set of initial conditions to train the algorithm, selecting conditions that reflect reality, including dry, wet, and snowy runway conditions and various initial landing speeds. We also incorporated brake degradation factors to demonstrate the algorithm's capability to adjust control inputs and regulate overall braking torque, maintaining efficient landing distances. Finally, we mapped these functionalities to the certification objectives recently defined by EASA.

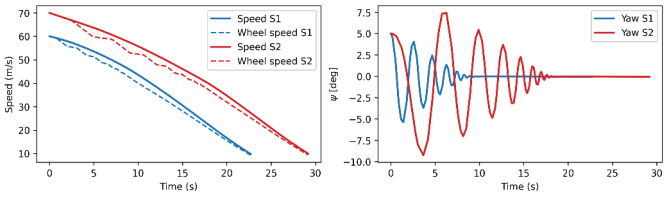

Figure 15. Wet runway, initial yaw, brake 1 at 80% performance, brake 2 at 90% performance

This work reveals promising results, showing that AI can be integrated into a safety-critical environment without modifying the main sensor and information system within the aircraft. The algorithm performs well under failure conditions, and its framework allows for the addition of new objectives as loss terms if needed. Furthermore, the algorithm does not require labeled data, as is necessary in supervised learning; introducing possible flight envelopes is sufficient to train the model. To increase the model's trustworthiness, higher fidelity aircraft models can be utilized. This can be further enhanced by surrogate modeling, which is our future work.

Acknowledgments

This work is part of the LANDOne project, funded by Innovate UK, a part of UK Research and Innovation, under grant number 10002411.

References

[1] A. Airbus, "Statistical analysis of commercial aviation accidents 1958-2019," *Blagnac: Airbus*, 2020.

[2] B. Ewers, J. Bordeneuve-Guibé, J.-P. Garcia, and J. Piquin, "Expert supervision of an anti-skid control system of a commercial aircraft," in *Proceedings of the 1996 IEEE International Symposium on Intelligent Control*. IEEE, 1998, pp. 420–425.

[3] Z. Jiao, D. Sun, Y. Shang, X. Liu, and S. Wu, "A high efficiency aircraft anti-skid brake control with runway identification," *Aerospace Science and Technology*, vol. 91, pp. 82–95, 2019.

[4] Z. Jiao, Z. Wang, D. Sun, X. Liu, Y. Shang, and S. Wu, "A novel aircraft anti-skid brake control method based on runway maximum friction tracking algorithm," *Aerospace Science and Technology*, vol. 110, p. 106482, 2021.

[5] L. D'Avico, M. Tanelli, S. Savaresi, M. Airoldi, and G. Rapicano, "A deceleration-based algorithm for anti-skid control of aircraft," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 168–14 173, 2017.

[6] L. D'Avico, M. Tanelli, and S. Savaresia, "Experimental validation of landing-gear dynamics for anti-skid control design," in *2018 European control conference (ECC)*. IEEE, 2018, pp. 2751–2756.

[7] D. Li, M. Lin, and T. Zhang, "Design and co-simulation of an antiskid brake system for a civil aircraft," *International Journal of Aeronautical and Space Sciences*, vol. 24, no. 3, pp. 845–852, 2023.

[8] S. Liu, Z. Yang, Z. Zhang, R. Jiang, T. Ren, Y. Jiang, S. Chen, and X. Zhang, "Application of deep reinforcement learning in reconfiguration control of aircraft anti-skid braking system," *Aerospace*, vol. 9, no. 10, p. 555, 2022.

[9] M.-B. Radac and R.-E. Precup, "Data-driven model-free slip control of anti-lock braking systems using reinforcement q-learning," *Neurocomputing*, vol. 275, pp. 317–329, 2018.

[10] P. Ioannou and Z. Xu, "Throttle and brake control systems for automatic vehicle following," *IVHS Journal*, vol. 1, no. 4, pp. 345–377, 1994.

[11] J. Tjonnas and T. A. Johansen, "Stabilization of automotive vehicles using active steering and adaptive brake control allocation," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 545–558, 2009.

[12] W. Xiang, P. C. Richardson, C. Zhao, and S. Mohammad, "Automobile brake-by-wire control system design and analysis," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 138–145, 2008.

[13] H. Liang, K. T. Chong, T. S. No, and S.-Y. Yi, "Vehicle longitudinal brake control using variable parameter sliding control," *Control Engineering Practice*, vol. 11, no. 4, pp. 403–411, 2003.

[14] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," *J. Sci. Comput.*, vol. 92, no. 3, sep 2022. [Online]. Available: https://doi.org/10.1007/s10915-022-01939-z

[15] U. Çelik and M. U. Demirezen, "Optimal reusable rocket landing guidance: A cutting-edge approach integrating scientific machine learning and enhanced neural networks," *IEEE Access*, 2024.

[16] G. Soudain, "Easa concept paper: guidance for level 1 2 machine learning applications," 2024.

[17] G.-. A. I. in Aviation, "Artificial intelligence in aeronautical systems: Statement of concerns," 2021.

[18] A. EASA, "C.: Formal methods use for learning assurance (formula)," Tech. Rep, Tech. Rep., 2023.

[19] C. Liu, D. Cofer, and D. Osipychev, "Verifying an aircraft collision avoidance neural network with marabou," in *NASA Formal Methods Symposium*. Springer, 2023, pp. 79–85.

[20] R. K. Schmidt, *The design of aircraft landing gear*. SAE International, 2021.

[21] P. D. Khapane, "Simulation of asymmetric landing and typical ground maneuvers for large transport aircraft," *Aerospace Science and Technology*, vol. 7, no. 8, pp. 611–619, 2003.

[22] A. Barnes and T. J. Yager, "Simulation of aircraft behaviour on and close to the ground." AGARD, 1985.

[23] Z. Ming, N. Hong, W. Xiao-hui, Q. Xiaomei, and Z. Enzhi, "Modeling and simulation of aircraft anti-skid braking and steering using co-simulation method," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 28, no. 6, pp. 1471–1488, 2009.

[24] S. Group, "Airbus a350-900 main landing gears," https://www.safran-group.com/products-services/airbus-a350-900-main-landing-gears, 2024, accessed: 2024-06-28.

[25] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle system dynamics*, vol. 21, no. S1, pp. 1–18, 1992.

[26] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, and A. J. Ramadhan, "Universal differential equations for scientific machine learning," *CoRR*, vol. abs/2001.04385, 2020. [Online]. Available: https://arxiv.org/abs/2001.04385

[27] M. Innes, A. Edelman, K. Fischer, C. Rackauckas, E. Saba, V. B. Shah, and W. Tebbutt, "A differentiable programming system to bridge machine learning and scientific computing," *CoRR*, vol. abs/1907.07587, 2019. [Online]. Available: http://arxiv.org/abs/1907.07587

[28] S. Omi, H.-S. Shin, N. Cho, and A. Tsourdos, "Dynamic deep-reinforcement-learning algorithm in partially observed markov decision processes," 2023.

A certifiable AI-based braking control framework for landing using scientific machine learning

Uzun, Mevlut

2024-09-29

Uzun M, Celik U, Guner G, et al., (2024) A certifiable AI-based braking control framework for landing using scientific machine learning. In: 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC), 29 September 2024 - 3 October 2024, San Diego, CA, USA
https://doi.org/10.1109/dasc62030.2024.10749078