DOI: 10.2507/34th.daaam.proceedings.xxx

ENHANCING REALISM IN ROBOTIC FLIGHT SIMULATORS: A CASE STUDY ON THE IMPLEMENTATION ANALYSIS OF A WASHOUT FILTER

Alina-Ioana Chira, Marian-Ciprian Bîlu, Răzvan-Ionuț Bălașa & Cătălin Iordache

This Publication has to be referred as: Chira, A[lina-Ioana]; Bîlu, M[arian] C[iprian]; Bălașa, R[ăzvan] I[onuț], & Iordache, C[ătălin] (2023). Enhancing realism in robotic flight simulators: a case study on the implementation analysis of a washout filter, Proceedings of the 34th DAAAM International Symposium, pp.xxxx-xxxx, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-xx-x, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/34th.daaam.proceedings.xxx

Abstract

The present work provides a comprehensive analysis of the implementation of a washout filter in an anthropomorphic-robotic flight simulator. It discusses the significance of adding a 7th degree of freedom (DOF) through a linear unit to enhance the pilot's perception within the flight simulator. The primary objective of this research was to develop a flight simulator on an anthropomorphic robot, serving as both a dynamic simulation system for pilot-in-the-loop applications and a pilot training platform. This simulator was designed to faithfully reproduce the sensations a pilot typically experiences during more demanding maneuvers, such as changes in G-forces in aircraft flight tests. A critical requirement for such a simulator is the accurate representation of its motion system, achieved through the proper implementation of the washout filter. This paper presents experimental results that validate the proposed solutions.

Keywords: robotics; flight simulation; washout filter; motion cueing; stability; 7th DoF.

1. Introduction

Nowadays, the aerospace industry places a high focus on safety procedures and equipment, leading to the development of increasingly durable and stable systems. While avionics systems themselves are becoming safer, the human factor is another element that can significantly impact the overall safety of a flight and should be taken into account. In recent years, aeronautics designs have become increasingly complex as the certification process has demanded higher levels of demonstration. Due to the increasing demands of the certification process, aeronautics designs have become more intricate in recent years. This is because certain systems in an aircraft are required to demonstrate failure rates of less than 10^{-9} flight hours over the operational life of the aircraft. To reduce the risks and costs of aircraft development ([1], [2]), flight simulation environments have been created to improve flying quality while maintaining a consistent level of operational performance. A summary of flight simulation history, as well as a list of applications and benefits of flight simulators, is presented [1]. As the aircraft design stages progress, more sophisticated and representative simulation environments are created [3]. Building on the above, the present research introduces a real-time flight simulator that enhances the fidelity of the effects of various aircraft dynamics on potential pilots and focuses training sessions on improving human performance. RoFSim (Dynamic Simulation System for pilot-in-the-loop applications – Robotic Flight Simulator) aims to design and implement a complex control system for simulating the overall dynamics of aerospace vehicles by utilizing a robotic manipulator arm [4] as a motion platform for use in motion perception test studies and studies to optimize training procedures for civilian and military pilots. The six degrees of freedom flight simulator comprises an industrial robotic arm on which the cockpit is placed, a central processing unit running the software, command instruments, and a Virtual Reality (VR) headset that serves as a display for the user (pilot). The architecture of RoFSim is designed to simulate, with high fidelity, changes in attitude and velocity when performing certain manoeuvres, helping the pilot understand the effects of their actions in-flight on a real aircraft. Furthermore, because the simulator's display is a VR headset, various aircraft models and weather condition settings for the training session can be chosen. Compared to static platform flight simulators, RoFSim is intended to be a versatile, low-cost tool designed to improve human performance.

RoFSim is a flight simulator mounted on an ABB 7600 robot arm [5] with six degrees of freedom and an additional 7th degree of freedom provided by a linear track, offering a distinct advantage over the Stewart platform. While it can move within a broader dynamic range, certain limitations still exist for safety reasons. These limitations stem from external factors, internal considerations, and safety requirements, all of which strictly restrict the joint angle ranges of the robot arm. To ensure the entire system can move in a specified direction without encountering nearby objects and to maintain the safety of the pilot and the entire setup, it becomes imperative to redefine the hardware and software limits for each joint of the robotic arm. This work outlines an approach to optimize the workspace of the robotic motion platform through three key aspects: hardware joint angle limitations, collision avoidance measures, and software-based joint angle restrictions. By applying these constraints, the robot's working space is carefully limited to create a secure and safe simulation environment.

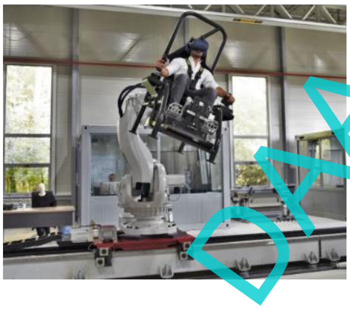

Fig. 1. RoFSim Flight Simulator

The paper's first section provides a description of the procedure for scaling the workspace of the six degrees of freedom simulator. It emphasizes three critical boundary conditions necessary for ensuring proper and safe simulator operation: hardware joint angle limitations, collision avoidance, and software-imposed joint angle restrictions. The primary focus of this work is the implementation of a washout filter (WF) on the RoFSim simulator, which translates aircraft dynamics into robot movements. One of the key components for achieving realistic motion is the software responsible for transforming aircraft accelerations into simulator motion cues. The objective is to enhance the simulator's realism and fidelity.

Experiments conducted with a simplified prototype of the flight simulator have revealed the limitations of the classical washout filter (CWF) architecture. Chapter 3 introduces the implemented motion algorithm for the RoFSim simulator, providing detailed explanations for each motion channel and specific system conditions.

Chapter 4 delves into the testing and validation scenarios of the system, encompassing flight plans, simulator configurations, and parameters extracted from the simulated aircraft dynamic model, all integrated with the defined CWF motion algorithm.

Finally, Chapter 5 presents the experimental data and results obtained from the testing phase of the RoFSim simulator.

2. Related work

As stated in the introduction, this study aims to develop, implement, and validate a washout filter algorithm that faithfully replicates the acceleration sensations a pilot experiences in a real aircraft, utilizing a motion platform with limited workspace.

The Washout filter, an essential motion algorithm, serves the critical function of translating real acceleration into the motion experienced within a flight simulator ([6], [7], [8]). Its core purpose is to provide motion cues that align with human perception [9] while adhering to the constraints and limitations inherent in the proposed flight simulator [10]. The significance of this filter is most pronounced in flight simulators employing the Stewart parallel robot mechanism [11]. While CWFs are renowned for their simplicity and adaptability [12], they have faced challenges in maintaining the fidelity of human perception. In recent years, new theories for implementing washout filters have gained prominence, with optimal ([14], [17]), adaptive ([7], [15], [16]), and robust ([13], [18]) theories emerging as pivotal areas of focus in this field.

An adaptive washout filter (AWF) was introduced by Nahin & Reid [19], building upon the CWF using a self-tuning algorithm. An optimal washout filter (OWF) is rooted in optimal theories [20], solving the Algebraic Riccati Equation (ARE) to meet specified requirements.

In our study, we also employ a classical washout filter, introducing fundamental changes to the required transfer functions. We leverage evolutionary algorithms to identify the optimal configuration within this framework. The primary objective is to minimize perception errors and align the simulator's actions more closely with those experienced in the actual workspace of a flight simulator.

Notably, one of the key innovations in our approach is the realistic representation of transient G-force losses/gains, achieved through the effective implementation of the translational channel of the classical washout filter. The adjustability of the robotic platform's movement constraints is detailed in the subsequent section.

3. RoFSim Workspace Scaling Procedure

To establish a suitable workspace for the RoFSim six-degree-of-freedom simulator, we defined hardware and software limitations. For safety reasons, we strictly constrained the robot's movement range to ensure the pilot's safety when using the simulator, preventing collisions with nearby objects. These adjustments resulted in additional constraints that reduced the robot's effective workspace. A comparison between the workspace of the IRB 7600 500/2.55 robot and the workspace when coupled with the RoFSim cabin reveals that the simulator's workspace is more restricted due to its larger dimensions.

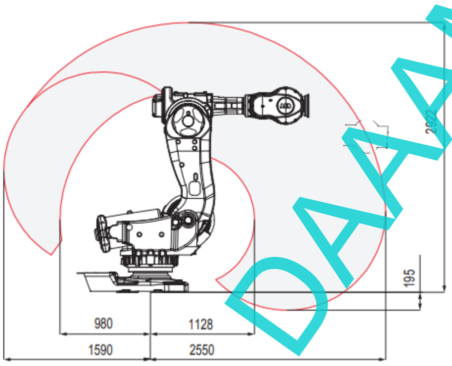

Fig. 2. Working Range IRB 7600-500/2.55 [5]

Depending on the specific application of the RoFSim simulator, different workspace constraints are established. It's important to note that there is a significant interdependency among joint angles, meaning that constraining one joint will affect the others. As a result, it is not possible to independently maximize the range of each joint angle. Consequently, we formulate the following optimization problem:

$$max\ \vec{\omega} \cdot \vec{f}_{kin}(\vec{q})\ ,\ \vec{q}\ \in\ R^6 \quad (1)$$

$$\vec{f}_{kin}(\vec{q}) = \{\Delta x, \Delta y, \Delta z, \Delta\phi, \Delta\theta, \Delta\psi\}^T \quad (2)$$

where x, y, z are the cartesian positions, ϕ, θ, ψ the rotation angles (roll, pitch, gyration), and $\vec{\omega} \in R^6$ the weighting factor vector.

In the process of formulating the workspace scaling procedure for the RoFSim system, the following essential conditions are considered: hardware limitations of joint angles, collision avoidance, and software-imposed limitations of joint angles.

A. Hardware Limitations of Joint Angles

The manufacturer provides specifications for the robot joint angle intervals in the data sheet. These hardware limitations are further complemented by software restrictions set by the manufacturer, which come into effect before reaching the hardware limits.

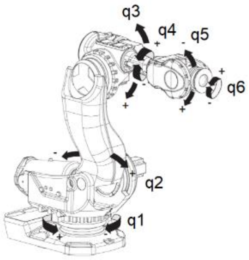

Fig. 3. Robot Motion Axes IRB 7600 - manufacturer configuration [5]

$$\vec{q} \in [\vec{q}_{min}, \vec{q}_{max}] \qquad (3)$$

IRB 7600 Robot	Axes	$\vec{\mathbf{q}}_{\mathbf{min}}$	$\vec{\mathbf{q}}_{\mathbf{max}}$	Speed limit	Acceleration limit
A	q_1	- 180 [°]	+ 180 [°]	75 [°/s]	-
B	q_2	-60 [°]	+85 [°]	50 [°/s]	-
C	q_3	-180 [°]	+60 [°]	55 [°/s]	-
D	q_4	- 300 [°]	+ 300 [°]	100 [°/s]	-
E	q_5	- 100 [°]	+100 [°]	100 [°/s]	-
F	q_6	- 360 [°]	+ 360 [°]	160 [°/s]	-
Track Gudel TMF- 4	q_7	-10 [mm]	20010 [mm]	90 m/min	86 [m/s^2]

Table 1. Defined IRB 7600 robot limits

B. Collision avoidance

The design of the RoFSim cabin significantly impacts the definition of joint motion ranges. When establishing collision avoidance criteria, we considered the following parameters: RoFSim cabin size, flange position, flange angle, and the dimensions of the subject (pilot). Safety must be ensured throughout the robot's entire range of motion to prevent collisions. The following objects were taken into account: the floor, the Gudel TMF-4 track, the IRB 4600 robot, the IRC5 controller of the IRB 4600 robot, the IRC5 controller of the IRB 7600 robot, and the nearby wall.

$$V(obiect_i, \vec{q}) \ \cap \ V(obiect_k, \vec{q}) \ = \ \emptyset \qquad (4)$$

where $obiect_i = \{cab\ RoFSim, robot\ IRB\ 7600, pilot, \dots\}$, and $obiect_k = \{robot\ IRB\ 4600, track\ Gudel\ TMF - 4, controller\ IRC5, wall, \dots\}$.

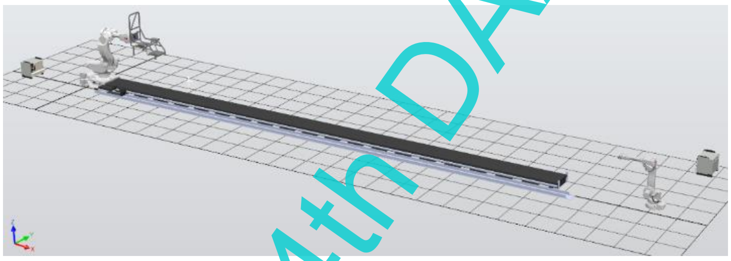

Fig. 4. RobotStudio Simulation - RoFSim and Objects in the Vicinity of the System

C. Software-Imposed Joint Angle Restrictions

Depending on the simulation type, various constraints and workspace limitations are applied. The following decisions were made following the analysis:

- Installation of SafeMove Hardware/Software Module: this module serves as an additional safety measure. If the IRB 7600 robot's path collides with surrounding objects (TMF-4 track, lab floor, IRC5 controller, or other areas/objects near the RoFSim workspace), it activates and blocks robot movement.
- Software Implementations - RoFSim Controller Module: to enhance safety during robot movements, the RoFSim Controller module has been programmed not to exceed +/- 90° (1.57 radians) on all three axes: X (Roll), Y (Pitch), and Z (Yaw). To prevent collisions with the Gudel track, the linear motion on these axes was constrained as follows: X (North-South) to +/- 3 meters, Y (East-West) to 0.4 meters, and Z (Vertical) to 0.4 meters. It's important to note that the simulator does not accept any motion commands while it is on the ground.
- Software Implementations in IRC5 Controller: additional modifications have been implemented to ensure that the q_5 axis does not exceed +/- 90° (1.57 radians).
- This constraint is necessary to avoid singularity errors during angular motions, and the q_5 axis must remain different from 0°.
- Singularities of the robotic arm occurs in any configuration when the wrist centre reaches the q_1 axis.

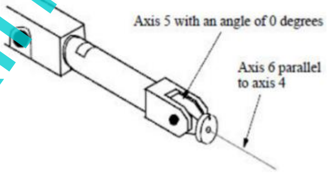

Fig. 5. Wrist Singularity [5]

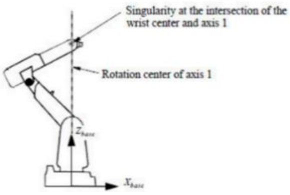
Singularity at the intersection of the wrist center and axis 1
Rotation center of axis 1
Z_base
X_base

Fig. 6. Arm Singularity [5]

Figure 7 illustrates the differences between the two working envelopes (2D format - white marking, and 3D format - green marking) with and without the RoFSim system coupled.

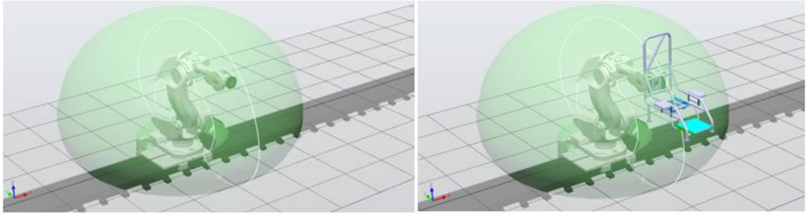

Fig. 7. IRB 7600 Robot Workspace Envelope: uncoupled (left) / coupled (right) – details from RobotStudio

Optimizing joint angle limits is a complex procedure as changes to one joint impact others. To simplify the process, we've divided the multidimensional optimization problem into several sub-problems. For each sub-problem, we've identified constraints and implemented measures to ensure simulator safety and integrity. This involved software modifications at various levels (RobotStudio/Rapid, Visual Studio/C# programming language) and hardware adjustments for each axis.

4. Implementation CWF

Flight simulators serve various purposes, including understanding human-machine interaction, studying pilot behaviour, and providing training. The effectiveness of a flight simulator largely depends on its ability to generate motion cues. By employing motion-cueing algorithms, we create a simulation environment.

The RoFSim's effectiveness is measured by its capability to generate realistic motion cues. Pilots in the cockpit should perceive motion cues similar to those experienced in a real aircraft during manoeuvres. However, due to the system's limited workspace, the RoFSim simulator cannot directly mimic aircraft motion. The platform would quickly reach its physical limits, rendering it unable to provide motion cues to the pilot.

To address this limitation, motion algorithms have been developed ([6] – [20]), including one tailored for the RoFSim simulator. This algorithm governs the generation of motion cues while keeping the robotic platform within its physical boundaries. The primary objectives of these motion algorithms are twofold: to provide lifelike motion cues to the pilot and to ensure the motion platform stays within its physical limits.

This chapter presents the motion algorithm designed specifically for RoFSim. It integrates mathematical models of vehicle dynamics and pilot behaviour into the overall system modeling framework.

The use of the robotic manipulator arm as a motion platform for closed-loop simulation raises two engineering challenges. Firstly, it requires kinematic schemes capable of replicating real-time, operator-commanded random system motions. Secondly, the entire system responsible for faithful motion replication must be adapted to fit within the specific motion envelope of the IRB 7600 robotic arm.

Central to the flexible and modular design of the proposed simulator is the control interface. This interface facilitates the connection between a realistic simulation environment, a simulation platform, and a virtual environment. Simultaneously, the control system supervises safety devices and provides manual and automatic control of the simulator. The scaling procedure developed in the previous chapter, aimed at assessing the user's perception of induced motion, is a critical aspect of the simulation system's development.

The motion reproduction algorithm's functionality involves filtering/scaling the ideal trajectory of the simulated vehicle. This trajectory is generated numerically based on the operator's commands. The algorithm's goal is to adapt the trajectory to fit within the motion platform's working space while providing a perception of motion as close as possible to that of a real vehicle.

The algorithm relies on the use of a smoothing filter or a combination of several smoothing filters. These filters are responsible for reproducing high-frequency motions and tilt coordination algorithms to replicate low-frequency motions. Essentially, vehicle motion is divided into low and high-frequency components. High-frequency components are replicated by physically moving the platform since they generate small, manageable displacements. In contrast, low-frequency components (linear accelerations) are not achieved by physically moving the platform but rather by utilizing the gravitational acceleration vector as a source of sustained acceleration.

The figure below provides a representation of the experimental motion perception system along with its specific subcomponents. The motion controller is responsible for transferring the pilot's commands to both the robotic motion platform and the visual system. Advanced control devices designed for the aircraft are utilized to transmit commands to the RoFSim simulation software module.

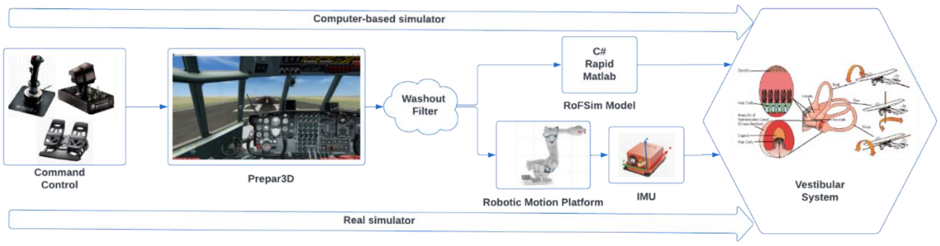

Fig. 8. Representation of the Experimental Motion Perception System Based on Subcomponents

Before delving into the detailed explanation of the implemented motion algorithm, it's crucial to establish a clear understanding of the relevant reference systems used in this study:

- Aircraft Reference System (S_A);
- RoFSim Simulator Reference System (S_S);
- Pilot - Aircraft Reference System (S_{AP});
- Simulator - Pilot Reference System (S_{AS}).

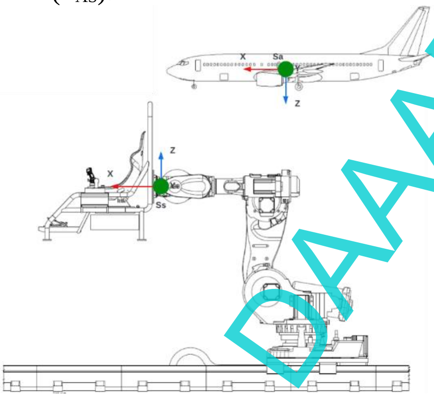

Fig. 9. RoFSim Reference Systems

When applying the motion algorithm, the transition from the aircraft reference system to the simulator reference system is accomplished using a matrix of directive cosines, employing a Z-Y-X rotation method.

$$\overrightarrow{S_A} = DCM_{ZYX} \cdot \overrightarrow{S_S} \tag{5}$$

where

$$DCM = \begin{bmatrix} \cos(\theta) \cdot \cos(\psi) & \cos(\theta) \cdot \sin(\psi) & -\sin(\theta) \\ sin(\phi) \cdot sin(\theta) \cdot cos(\psi) - \cos(\phi) \cdot \sin(\psi) & sin(\phi) \cdot sin(\theta) \cdot sin(\psi) + \cos(\phi) \cdot \cos(\psi) & sin(\phi) \cdot cos(\theta) \\ cos(\phi) \cdot sin(\theta) \cdot cos(\psi) + \sin(\phi) \cdot \sin(\psi) & cos(\phi) \cdot sin(\theta) \cdot sin(\psi) - \sin(\phi) \cdot \cos(\psi) & cos(\phi) \cdot cos(\theta) \end{bmatrix} \tag{6}$$

Within the operation of the motion algorithm, we applied both constraints and scaling to the input signals for the translational and rotational channels. This constraint and scaling uniformly change the input signal's amplitude across all frequencies. Limiting, a non-linear process, modifies the signal to restrict it to a specific magnitude, ultimately reducing the motion response of the simulator.

In our approach, we used a third-order polynomial for scaling, which was integrated into the general motion algorithm scheme. Let x represent the input, y the output, x_{max} the desired maximum input, y_{max} the maximum output, and s_0 and s_1 the slopes at $x = 0$ and $x = x_{max}$ respectively.

The third-order polynomial used to achieve specific characteristics is in the following form:

$$y = c_3x^3 + c_2x^2 + c_1x^1 + c_0 \tag{7}$$

Parameter scaling for the translational channel:

$$max\ input\ (x_{max}) = 6\ m/s^2 \tag{8}$$

$$max\ output\ (y_{max}) = 6\ m/s^2, \text{ for } X \tag{9}$$

$$max\ output\ (y_{max}) = 0.8\ m/s^2, \text{ for } Y \text{ and } Z \tag{10}$$

and the coefficients are:

$$\begin{gathered} s_0 = 1.5, \text{ for X} \\ s_0 = 0.2, \text{ for Y and Z} \\ s_1 = 0, \text{ for X, Y and Z} \end{gathered} \tag{11}$$

Parameter scaling for the rotational channel:

$$max\ input\ (x_{max}) = 3.14 \text{ rad/s} \tag{12}$$

$$\text{max output } (\text{y}_{\text{max}}) = 1.57 \text{ rad/s} \tag{13}$$

and the coefficients are:

$$\begin{gathered} s_0 = 0.785, \text{ for X, Y and Z} \\ s_1 = 0, \text{ for X, Y and Z} \end{gathered} \tag{14}$$

Linear accelerations are limited to 6 m/s² on all axes for the translational channel, and the limitation for the rotational channel is 3.14 rad/s.

The structure of the aircraft simulation system, based on a robotic platform motion system, is depicted in the figure below. Operator (pilot) inputs are first directed to the aircraft dynamic model, which generates the aircraft state vector. The aircraft state vector is then processed through the motion algorithm, producing the desired motion cues and the states of the robotic platform that provide the motion to the simulator.

Subsequently, the desired states of the robot platform are transformed from the degree of freedom space to the axis workspace, generating commands for the six robot axes. These axis motion commands are used as input for the robotic platform's dynamics, resulting in the actual motion of the simulator.

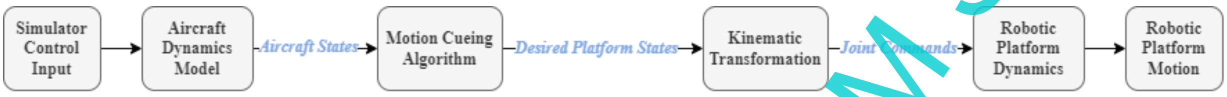

Fig. 10. RoFSim simulator structure

The primary concept of the CWF is to replicate specific forces and rotations, simulating sensations similar to those experienced in a real cockpit. The filter's quality is assessed by minimizing the „feel" error, which reflects the difference between the specific forces and rotations perceived from the simulator and those of the aircraft. The CWF algorithm includes three channels: translational, rotational, and tilt coordination, as shown in Figure 11.

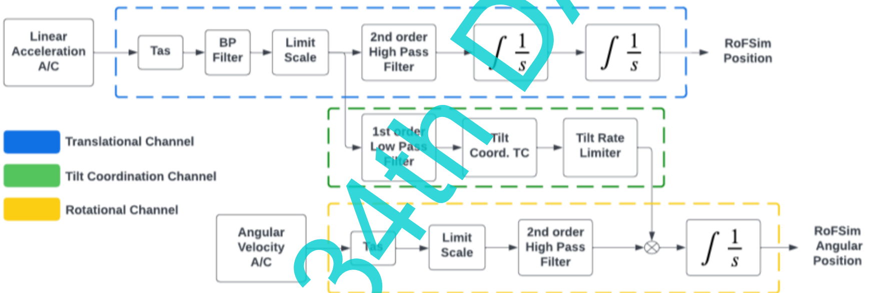

Fig. 11. Motion Perception Structure for RoFSim simulator

A. Translational Channel

The translational channel manages linear acceleration by transitioning from the aircraft reference system (S_A) to the RoFSim simulator reference system (S_S). A band-pass filter is applied, where values less than a user-defined threshold (0.06) are set to zero. This is followed by the limiting and scaling phase, explained in the previous chapter, ensuring RoFSim simulator movement accuracy to within 1 mm.

To address the robotic platform's inability to replicate low-frequency linear accelerations, a high-pass filter is used. The filter takes the form of:

$$y = \alpha \cdot (y_o + x - x_0) \tag{15}$$

where: y is current output, $\alpha = 0.95$, y_o is previous output, x is current input, x_0 is previous input.

When the acceleration input remains constant for a period, the robotic platform gradually returns to the zero position, providing a stable starting point for the next acceleration. The subsequent step involves double integrating the filtered accelerations to calculate the position in three directions: *X*, *Y*, and *Z* of the RoFSim simulator:

$$s = s_o + v_o \cdot t + a \cdot \frac{t^2}{2} \tag{16}$$

where: s is final position, s_o is initial position, v_o is initial velocity, a is acceleration and t – time.

B. Tilt Coordination Channel

The Tilt Coordination channel converts accelerations into Euler angles, allowing the pilot to perceive gravitational force as translational acceleration. This transformation is achieved through a transfer function that switches from accelerations to angles.

$$\phi = arcsin\left(\frac{x}{g}\right), \text{ where } g = 9.803\,\frac{m}{s^2} \tag{17}$$

$$\theta = arcsin\left(\frac{-Y}{g \cdot cos\theta}\right)$$

$$\psi = 0$$

where ϕ, θ, ψ represent roll, pitch and yaw.
Input received from the translational channel is passed through a low pass filter:

$$y = \ y_o + \alpha \cdot (x - x_0) \qquad (18)$$

where: y is current output, α = 0.05, y_o is previous output, x is current input and x_0 represents previous output.
A rate limit of 0.2 rad/s is applied to restrict rotational speeds. This ensures that the pilot does not perceive rotation but only experiences the gravitational force decomposed into the two axes (roll and pitch), allowing the pilot to feel linear accelerations.

$$\begin{aligned} & if\ |x_1 - x_0| > limit \\ & \quad if\ x_0 < x_1 \rightarrow\ x_0 =\ x_0 +\ limit \\ & \quad else\ x_0 = x_0 - limit \\ & \quad \mathrm{else\ x_0 = x_1} \end{aligned} \qquad (19)$$

where: $limit$ = 0.2, x is current input and x_0 represents previous input.

C. Rotational Channel

The rotational channel is directly responsible for the rotational movements of the robotic platform. This channel takes the angular velocities of the aircraft as its primary input and the Euler angles from the tilt coordination channel as its secondary input.
In the first phase, the input is limited and scaled as described above. Then, a high-pass filter, identical to the one used for the translational channel, is applied. The resulting output after filtering is summed with the output received from the tilt coordination channel.
The final step involves integrating the angular velocities to obtain the angular position of the RoFSim simulator.
The CWF algorithm described above was implemented in the C# programming language. In the graphical user interface of the RoFSim Controller module, both inputs (linear accelerations, angular velocities) and outputs (*X*, *Y*, *Z* position, and angular position of the RoFSim simulator) are displayed in real-time in graphical format.

5. Materials and methods

The flight plan for the final testing of the CWF algorithm includes taking off from Henri Coandă Otopeni International Airport - LROP, runway 08R, following the VFR flight procedure, with a climb to 2500 ft, and then making a 90° turn with a compass heading of 170.
During the testing and validation phase of the CWF algorithm, the MTi-G-710-2A8G4 inertial measurement system (IMU) was positioned in the subject's (pilot's) head area. The collected data include linear accelerations and Euler angles. Various parameters were recorded and are presented in Tables 2 to 4.

Parameters IMU	**Unit**
Time	UTC
Acceleration X	m/s^2
Acceleration Y	m/s^2
Acceleration Z	m/s^2
Roll	m/s^2
Pitch	deg
Yaw	deg

Table 2. Parameters recorded by the IMU system

Parameters A/C Model	**Unit**
Time	UTC
Pitch	deg
Roll	deg
Acceleration world X	m/s^2
Acceleration world Y	m/s^2
Acceleration world Z	m/s^2
Velocity Body X	feet/s
Velocity Body Y	feet/s
Velocity Body Z	feet/s

Acceleration Body X	m/s^2
Acceleration Body Y	m/s^2
Acceleration Body Z	m/s^2
Plane Latitude	deg
Plane Longitude	deg
Plane Altitude	deg
Plane Heading Degrees True	deg
Incidence Alpha	deg
Incidence Beta	deg
Airspeed Mach	Mach
Airspeed True	Knots
Airspeed True Calibrate	Knots
Max G Force	G Force
G Force	G Force
Aileron Average Deflection	rad
Elevator Deflection	rad
Rudder Deflection	rad

Table 3. Parameters recorded from the dynamic model of the simulated aircraft

Parameters CWF	Unit
Time	UTC
Input Acceleration X	m/s^2
Input Acceleration Y	m/s^2
Input Acceleration Z	m/s^2
Robot Input Acceleration X	m/s^2
Robot Input Acceleration Y	m/s^2
Robot Input Acceleration Z	m/s^2
Robot Output Position X	m
Robot Output Position Y	m
Robot Output Position Z	m
Input Rotational Velocity X	rad/s
Input Rotational Velocity Y	rad/s
Input Rotational Velocity Z	rad/s
Robot Input Rotational Velocity X	rad/s
Robot Input Rotational Velocity Y	rad/s
Robot Input Rotational Velocity Z	rad/s
Robot Output Angle X	rad
Robot Output Angle Y	rad
Robot Output Angle Z	rad

Table 4. Parameters recorded from the CWF algorithm

6. Results

The test scenario was conducted over 20 flight sessions to validate the functionality of the CWF motion algorithm governing the RoFSim simulator's motion. Data from test number 8 were used to graphically display the simulation results.

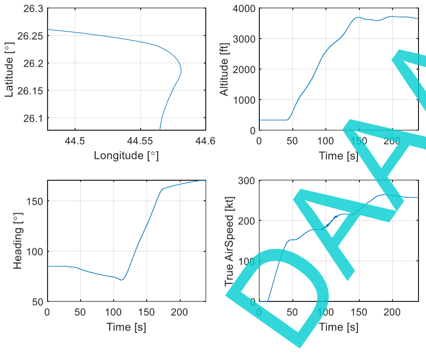

Fig. 12. Lat-Long; Heading, Altitude, TAS vs time

In order to validate the implemented motion algorithm, the linear accelerations of the robot and those recorded by the IMU system were compared, as well as the Euler angles (roll, pitch and yaw).

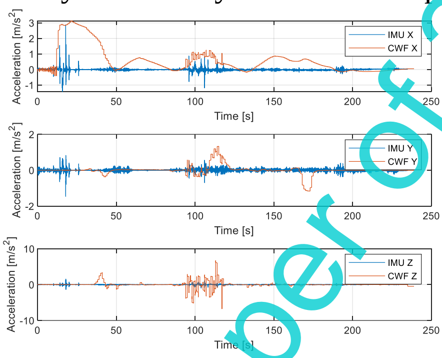

Fig. 13. Comparison of Linear Accelerations Recorded by IMU and CWF algorithm

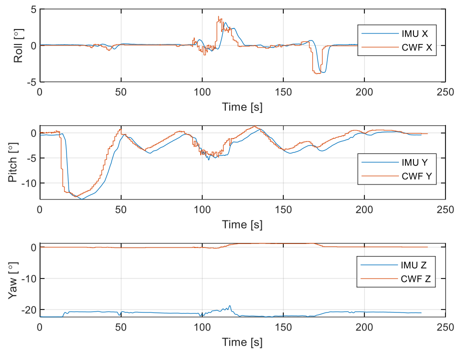

Fig. 14. Comparation of Euler Angles Recorded by IMU and CWF algorithm

7. Conclusion

In this paper, we implemented a washout filter in the RoFSim simulator to replicate sensations similar to those experienced by a pilot during real flight.

Firstly, we devised a workspace scaling procedure for the RoFSim simulator to ensure it possesses an adequate workspace. We approached this as an optimization problem with three primary conditions: hardware joint angle constraints, collision avoidance, and software-imposed joint angle restrictions. For each of these issues, we identified constraints and implemented appropriate measures to ensure the simulator's safety and integrity. We made modifications at multiple levels, including software adjustments using RobotStudio and programming in C#/Rapid languages. These modifications underwent rigorous testing and validation through both offline and real-time simulations.

Additionally, we developed and implemented a motion algorithm aimed at providing realistic motion feedback to the pilot within the cockpit while adhering to the specified limits of the robotic motion platform. This motion algorithm, referred to as the Classical Washout Filter, underwent testing in a dedicated campaign. Its functionality was confirmed as it successfully replicated the motion of the RoFSim simulator within the confines of its workspace, thereby offering the pilot a faithful perception of motion similar to what is experienced in a real cockpit. To validate the motion algorithm, a total of 20 flights were conducted in the test campaign. This validation process involved comparing the data recorded by the IMU system with the parameters recorded by the motion algorithm. Graphical representations of the results clearly demonstrate that the simulator behaves accurately, and the motion algorithm effectively conveys motion cues to the pilot, closely resembling those experienced in a real cockpit. The extensive series of tests performed underscores the stability of the RoFSim simulator and confirms the effectiveness of the developed configuration.
As part of our future work, we aim to enhance the motion algorithm model to achieve a higher fidelity in motion indices by implementing an optimal algorithm.

8. Acknowledgements

The work has been funded by the Romanian Government through the Ministry of Research, Innovation and Digitisation (MCID), under the “AEROEXPERT 2019-2022” program, financial agreement no. 8N/2019.

9. References

[1] Brain, C.J.; Clayton, T.D.; Ward, N.J. (1996). Organizing safety-related aircraft simulation. Presented at: 2nd Test and Evaluation International Aerospace Forum; London, UK, DOI: 10.2514/6.1996-3331
[2] Allerton, D.J. (2010). The impact of flight simulation in aerospace. The Aeron Journal 114(1162):747-756
[3] Mendonça, C.B.; Silva ET, Curvo M, Trabasso LG (2013). Model-based flight testing. Journal of Aircraft 50(1):176-186, DOI: 10.2514/1.C031778
[4] Afloare, A.I.; Apostolescu, N.; Chira, A.I.; Munteanu, C.E. (2019). Inverse Kinematic Solution of a 6 DoF Serial Manipulator, Proceedings of the 30th DAAAM International Symposium, pp.0628-0631, B. Katalinic (Ed.), Published by DAAAM International, ISSN 1726-9679, Vienna, Austria
[5] ABB Robotics - Product specification IRB 7600,
Available from: https://new.abb.com/products/robotics/robots/articulated-robots/irb-7600, Accessed: 2023-10-02
[6] Grant, P. R.; Reid, L. D. (1997). Motion washout filter tuning: Rules and requirements, Journal of Aircraft, 34(2), 145–151, DOI: 10.2514/2.2158
[7] Parrish, R. V.; Dieudonne, J. E.; Bowles, R. L.; Martin Jr, D. J. (1975). Coordinated adaptive washout for motion simulators, Journal of Aircraft, 12(1), 44–50, DOI: 10.2514/3.59800
[8] Sivan, R.; Ish-Shalom, J.; Huang, J. (1982). An optimal control approach to the design of moving flight simulators. IEEE Transactions on Systems, Man, and Cybernetics, 12(6), 818– 827, DOI: 10.1109/TSMC.1982.4308915
[9] Song, C.C.; Liaw, D.C.; Chung, W.C. (2002). Washout-filter based bifurcation control of longitudinal flight dynamics. IEEE Region 10 Conference on Computers, Communications, Control and Power (pp.1646–1649), Beijing, China
[10] Wang, S. C.; Fu, L. C. (2004). Predictive washout filter design for VR-based motion simulator, IEEE International Conference on Systems, Man and Cybernetics, SMC 2004, 6291–6295, The Hague, Netherlands
[11] Affan, M.; Ahmed, S. U.; Manek, A. I.; Uddin, R. (2019). Design and implementation of the washout filter for the Stewart-Gough Motion Platform, 2019 International Conference on Computational Intelligence and Knowledge Economy, (pp. 415–419), Dubai, UAE
[12] Liao, C.S.; Huang, C.F.; Chieng, W.H. (2004). A novel washout filter design for a six degree-of-freedom motion simulator, JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing, 47(2), 626–636
[13] Asadi, H.; Mohamed, S.; Lim, C. P.; Nahavandi, S. (2016). Robust optimal motion cueing algorithm based on the linear quadratic regulator method and a genetic algorithm, IEEE Transactions on Systems, Man, and Cybernetics: Systems, PP (99), 1–17
[14] Gharib, M. R. (2020). Comparison of robust optimal QFT controller with TFC and MFC controller in a multi-input multioutput system, Reports in Mechanical Engineering, 1(1), 151– 161
[15] Huang, C.; Fu, L. (2006). Human Vestibular Based (HVB) Senseless maneuver optimal washout filter design for VR-based motion simulator, IEEE International Conference on Systems, Man and Cybernetics, pp. 4451–4458, Taipei, Taiwan
[18] Moaveniah, M.; Gharib, M. R.; Daneshvar, A.; Alimardani, S. (2011). Control of human hand considering uncertainties, International Conference on Advanced Mechatronic Systems, pp. 17–22, IEEE
[19] Nahon, M. A., Reid, L. D., & Kirdeikis, J. (1992). Adaptive simulator motion software with supervisory control. Journal of Guidance, Control, and Dynamics, 15(2), 376–383
[20] Telban, R., Cardullo, F., & Houck, J. A. (2002). Nonlinear, human-centered approach to motion cueing with a neurocomputing solver. In AIAA Modeling and Simulation Technologies Conference and Exhibit, pp.1–10, Monterey, CA, USA