Fault Tolerance Tool for Human and Machine Interaction & Application to Civilian Aircraft

Javier Viaña and Kelly Cohen

February 26, 2021

2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Guayaquil, Ecuador, 2019, pp. 1-2, doi: 10.1109/LA-CCI47412.2019.9037045 (Preprint and Prereview Version)

Fault Tolerance Tool for Human and Machine Interaction & Application to Civilian Aircraft

Javier Viaña
Department of the Aerospace Engineering & Engineering Mechanics
University of Cincinnati
Cincinnati, USA
vianajr@mail.uc.edu

Kelly Cohen
Department of the Aerospace Engineering & Engineering Mechanics
University of Cincinnati
Cincinnati, USA
cohenky@ucmail.uc.edu

***Abstract*— Enhancing human-machine interaction is critical to aerospace applications. An essential requirement in safety critical systems is the clear need to guarantee trustworthiness of a system as well as V&V (Verification and Validation). However, the current state of the art concerning decision support systems lacks effective tools in this area. The Coherence Function Package, introduced in this research, is a tool towards providing assurance that the action needed has the approval of both the human and the machine in terms of SAFETY. These algorithms shed light on the future of an Explainable Artificial Intelligence (XAI, [1]), that fosters a synergy between these two factors. This vital requirement that has been further underscored after the tragic events of the Boeing 737 Max 8 crashes [2]. Preliminary results show that the proposed approach is not only able to detect any errors in the system, it also assists in circumventing conflicts leading to incoherence and suggests a preferred solution in real-time.**

Keywords*— *Human-Machine Interaction, Flight Critical Systems, Aircraft Safety, Tautology, Implications, Equivalences, Recursive Functions.

I. Introduction

A. Problem

Human-machine interaction malfunctions represent a paradigm of catastrophic failure in Typical Civilian Aircrafts (TCAs).

B. Main Purpose

Offer an effective approach to address this class of problems using AI to improve trust and overall fault tolerance.

C. Approach

A tool that combines the information of both user and software to resolve any possible errors in the interaction. The algorithms of this package use the principles of tautologies to make relational chains that determine the veracity of all system inputs. This technique has been benchmarked with a simple yet realistic scenario to test the impact that it's integration in real life would have on the selected system. It has been chosen to examine the applicability of the proposed approach to a TCA.

II. Related Work

Artificial Intelligence (AI) has previously been used for handling unexpected circumstances in the Guidance Navigation and Control (GNC) of the aircraft [3]. Moreover, fully developed Automated Human-Machine Interfaces (HMIs) are currently integrated on-board many off-the-shelf products. In [4], a special application of HMIs is considered for the Remotely Piloted Aircraft System (RPAS). Similarly, the use of Cognitive HMI for Single Pilot Operations (SPOs) is discussed in [5]. There are various the advancements done in the field of coherence and verification of decision making. In [6] Symbolic Model Checkers (SMCs) are used to verify the transition relationships that define the model. If a certain property is not correct according to the SMC, the system will return a counterexample and provide the necessary information about the failure.

III. Theoretical Background

In the present study, all statements are identified with capital letters. In the same way, an implication involves the presence of at least two statements, cause and effect. A possible straight forward implication for the case of the TCA could be the following: "Anti-stall system specification: Nose Up $\rightarrow$ Action to take: Nose Up"

The use of "→" refers to an implication, whereas the use of "↔" expresses an equivalence, or so to speak a double-sided implication. Identifying all the individual statements and the explicit implications of a complex system like the TCA can be cumbersome, especially in those that have a large number of degrees of freedom. But it is precisely in those systems in which this Coherence Package or any AI plays a key role, where the Classic Control Theory is no longer applicable, and it is needed a much more elaborated network of implications and equivalences. The core algorithm of the software is based in the use of four simple tautologies, shown below. A tautology is so to speak an unconditional truth, a theorem, or a pillar on which all the code is based. For the reference, the use of $\bar{A}$ expresses the opposite action/statement to the one specified by A.

$$(A \rightarrow B) \wedge (B \rightarrow C) \rightarrow (A \rightarrow C) \tag{1}$$

$$(A \rightarrow B) \wedge (B \leftrightarrow C) \rightarrow (A \rightarrow C) \tag{2}$$

$$(A \leftrightarrow B) \wedge (B \rightarrow C) \rightarrow (A \rightarrow C) \tag{3}$$

$$(A \leftrightarrow B) \leftrightarrow (\bar{A} \leftrightarrow \bar{B}) \tag{4}$$

IV. METHODOLOGY

The Coherence Package uses a learning process to create all the implicit information that is hidden inside a certain statement. To differentiate between the input introduced in the system and the learned-input, or extended version of this, the word “ext” is used. Hence, a matrix called $\boldsymbol{A}$ collects all the implications specified in the original statement (explicit sub-statements), and a different matrix $\boldsymbol{A}_{ext}$ complements the former by using the four tautologies mentioned, obtaining thus the extended version of $\boldsymbol{A}$ (explicit and implicit sub-statements). In the same way the matrix $\boldsymbol{B}$ contains the equivalences shown in the original statement, and $\boldsymbol{B}_{ext}$ the deduced equivalences. The programming that lies behind this software involved the use of several layers to gradually approach the problem. Perhaps the highest difficulty of such coding was the use of the recursive functions needed to verify the current status of the evolution of the learning process. But after obtaining both $\boldsymbol{A}_{ext}$ and $\boldsymbol{B}_{ext}$, the truth or coherence of the original statement can be validated immediately. Resulting into a simple problem of spotting any misalignments in the decision.

V. PRELIMINARY RESULTS

Currently, the computer science community strives to search for an upgrade of the current methods for the Guidance Navigation and Control of complex aircrafts. The GNC Logic of a TCA posed a challenging scenario to test the algorithms. The data for the realistic scenario considered was extracted from [7]. For the testing, a reduced problem of 24 statements was considered for the display of results, and a more complex one of 108 statements to validate performance. In the next figures, the input implications and individual statements have been plotted on the left. Finally, on the right representation of the figure, it can be observed how the derived (implicit and explicit) implications and the combination of statements leads to an incoherence, since several statements and their opposites are taking place.

The Coherence Package was not only able to spot the errors, but it did also correct them immediately for both the simple and complex cases considered. Such correction implied suppressing the minimum amount of linkages so that the resulting new version of primal implications are coherent. A result that remarks even more the importance of this software, since its integration in an on-board computer could perhaps make a major change in the outcome of a possible HMI malfunction. The case studied is set apart from regular AI, since it has the value of reliability and fault tolerance. The ability to determine how vulnerable a certain system. When selecting an example of a system with no sensing redundancies the algorithm spots immediately such weakness and addresses it. Thus, it serves as a tool to demonstrate and analyze the robustness and sensitivity of a certain machine. Where any fragile linkage or uncertainty is corrected before considering coherent and acceptable any action. Having in the end not only the detection capability but also the “intelligence” to determine the best option according to the given inputs.

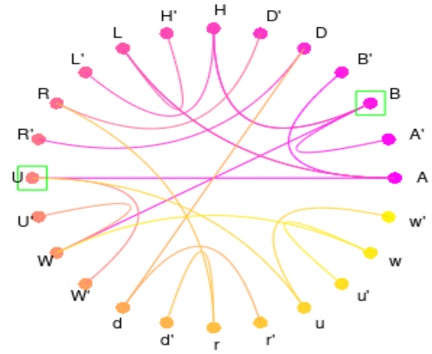

Fig. 1. Implications of the Statement (Matrix A).

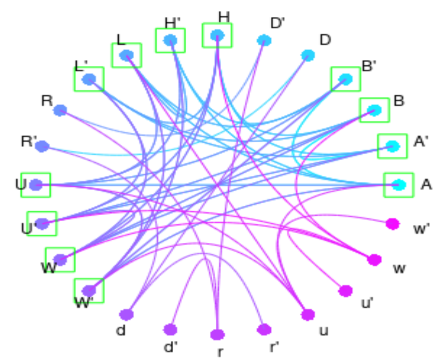

Fig. 2. Derived Implications (Extended Matrix A).

REFERENCES

[1] S. Kaushik, “Holy Grail of AI for Enterprise - Explainable AI,” https://www.kdnuggets.com/2018/10/enterprise-explainable-ai.html

[2] G. Travis, “How the Boeing 737 Max Disaster Looks to a Software Developer,” IEEE, https://spectrum.ieee.org/aerospace/aviation/how-the-boeing-737-max-disaster-looks-to-a-software-developer

[3] K. Button, “A.I. in the Cockpit,” Aerospace America, https://aerospaceamerica.aiaa.org/features/a-i-in-the-cockpit/

[4] Y. Lim, A. Gardi, R. Sabatini, S. Ramasamy, T. Kistan, N. Ezer, J. Vince, R. Bolia, “Avionics Human-Machine Interfaces and Interactions for Manned and Unmanned Aircraft,” Progress in Aerospace Science, https://www.researchgate.net/publication/326817849_Avionics_Human-Machine_Interfaces_and_Interactions_for_Manned_and_Unmanned_Aircraft

[5] M. Gearhart, “Human Factors and the Road to Single Pilot Operations,” Bridgewater State University, https://vc.bridgew.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1276&context=honors_proj

[6] T. J. Arnett, “Verification of Genetic Fuzzy Systems,” University of Cincinnati, https://etd.ohiolink.edu/!etd.send_file?accession=ucin1460731645&disposition=inline

[7] K. P. King and A. McPherson, “A Piloted Simulator Study of a Jet VTOL Aircraft in Partially Jet-Borne Flight,” Ministry of Aviation Supply, Aeronautical Research Council Reports and Memoranda, http://naca.central.cranfield.ac.uk/reports/arc/rm/3647.pdf