DOI: 10.13009/EUCASS2023-849
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

Concepts for Independent Monitoring of Flight Control Laws

*Dominik Hübener**†, Robert Luckner* and Guido Weber***
** Technische Universität Berlin*
Marchstraße 12-14, 10587 Berlin, Germany
*** Liebherr-Aerospace Lindenberg GmbH*
Pfänderstraße 50-52, 88161 Lindenberg im Allgäu, Germany

d.huebener@tu-berlin.de – robert.luckner@tu-berlin.de – guido.weber@liebherr.com
† Corresponding Author

Abstract
Electronic flight control systems are safety critical systems requiring highest integrity and availability levels. Therefore, the development of flight control law software follows rigorous processes. However, complete absence of development errors in the flight control law cannot be guaranteed. Such errors would represent a single point of failure that can lead to flight control system failure. Independent monitoring of the flight control function is a potential solution.
This paper investigates concepts for Independent Monitors. The scope of the monitor is defined, advantages of potential monitoring levels are discussed, suitable detection measures are explained, and two concepts for Independent Monitors are proposed.

1. Introduction

Electronic flight control systems (FCS) are safety critical systems requiring highest levels of integrity and availability. Today's flight control laws (FCL) and the embedded FCL software are highly complex. It is generally not practical (and may not even be feasible) to establish a development process that can conclusively demonstrate the absence of development errors in the FCL. As new actors push into the market, e.g. electric Vertical Take-Off and Landing aircraft (eVTOLs) for Urban Air Mobility, and as new functions further increase the complexity of FCL, the risk of latent FCL development errors rises [1], [2]. Means to mitigate the effects of such errors are becoming more and more important. This paper investigates possible concepts for an Independent Monitor of Flight Control Laws to detect and mitigate the effects of FCL development errors.

1.1 Motivation

Commonly, development assurance is used to mitigate the risk of development errors. AMC 25.1309 [3] defines a *development error* as "a mistake in requirements, design, or implementation". Development assurance is a process-based technique used to limit the likelihood of development errors that could impact aircraft safety. SAE ARP4754A [4] provides guidelines for the processes used to develop civil aircraft and airborne systems.

Additionally, architectural means are applied to limit the consequences of development errors. State-of-the-art FCS architectures often compare the outputs of redundant lanes with similar FCL software [5], [6], [7], [8], [9]. Dissimilarity is sometimes implemented on code level, to mitigate the effects of software coding (implementation) errors. However, nearly all serious accidents (of different industry sectors) in which software was involved are related to requirements flaws and not coding errors [10]. Therefore, the FCL software can be the source of common mode errors[1] and failures.

[1] An error which affects a number of elements otherwise considered to be independent (ARP4754A § 2.2) [4].

DOI: 10.13009/EUCASS2023-849

A *failure* is an occurrence which affects the operation of a system or component, such that it can no longer function as intended. This includes loss of function and malfunction [3].

The European Union Aviation Safety Agency (EASA) has released a generic certification review item (CRI), on the subject of common mode failures and errors in flight control functions. The EASA CRI [11] highlights that development assurance alone is not necessarily sufficient to establish an acceptable level of safety for flight control functions. Additional "mitigation means or techniques should be provided to protect against Common Mode Failures and Errors." Also, the EASA recognizes that "dissimilarity in the high-level specification of Flight Control Laws cannot be easily implemented" and, that "monitoring of the Flight Control Laws may be a possible mitigation against common mode errors" [11].

1.2 Objective

The objective of the research project MODULAR, funded by the German Federal Ministry of Economic Affairs and Climate Action, is to investigate an Independent Monitor as a possible mitigation means against FCL development errors.

An early detection and triggering of an appropriate response, e.g. switching to an alternative FCL, is key to counteract FCL development errors and the resulting FCS failure. It is assumed that an alternative exists that is so simple that it is considered error-free, e.g. direct mode FCL. In case of a failure detection, the FCS can switch to the simpler FCL, thus avoiding loss of function. Here, the focus lies on the failure detection, while reconfiguration methods are out of scope of this paper.

The objective of the Independent Monitor is to increase the safety of the FCS while maintaining highest rates of availability. Leveson states that most safety problems in software of complex, safety critical systems (from various industries) stem from requirement flaws and not coding errors [10]. To avoid common mode requirement errors, the Independent Monitor shall be functionally independent from the normal mode FCL. Also, the Independent Monitor shall detect failures, i.e. erroneous function (malfunction), of the FCS caused by FCL development errors.

To maintain availability of the FCS in the Normal Mode, the Independent Monitor shall only detect failure conditions that are classified as hazardous or catastrophic. Additionally, it shall be robust against false detections under foreseeable operational conditions to avoid spurious detection of failures that reduce the availability of the FCS. The Independent Monitor shall be simple to avoid the introduction of new development errors. Therefore, the level of complexity of the Independent Monitor shall be significantly lower than normal mode FCL to be monitored.

1.3 Content

This paper investigates concepts for Independent Monitors. The scope of the FCL to be monitored is defined, advantages of potential monitoring levels are discussed, suitable detection measures are explained, and two concepts for Independent Monitors are described.

2. Scope of Monitor, Monitoring Levels and Failure Detection Measures

The first question that needs to be answered is, what is the scope of the FCL to be monitored. The second question to be answered is, what monitoring level is suitable to achieve the design objectives of the Independent Monitor. And at last, what detection measures can be applied to detect FCS failures. This shall be discussed in the next subsections before two concepts for an Independent Monitor are proposed in Section 3.

2.1 Scope of the Independent Monitor for Flight Control Laws

Figure 1 shows a part of a schematic FCS architecture. The FCS consists of the pilot control devices (sidesticks, rudder pedals, thrust levers, etc.), the Flight Control Computer (FCC), the control surfaces and the cockpit displays.

DOI: 10.13009/EUCASS2023-849

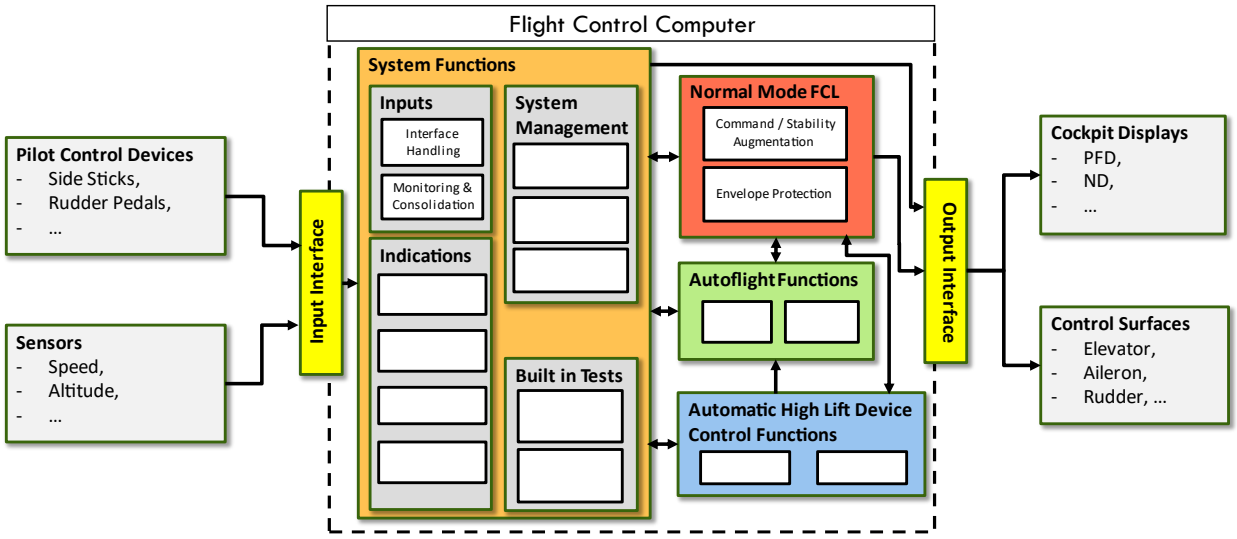

Figure 1: Schematic architecture of a Flight Control System (adapted from [12]).

Several functions are implemented on the FCC. Here however, the focus is on the flight control functions: the *Input Monitoring & Consolidation function (IM/C)* in the block *"Inputs"*, and the *normal mode FCL* that commands the control surface deflections.

The IM/C comprises the monitoring sub-function that checks the received input signals for correct range, refresh rate, parity, integrity and monitors the redundant signals for deviations from each other. The consolidation sub-function uses the redundant signals for forming consolidated internal parameters (e.g. by voting algorithms) for usage in the FCL.

The normal mode FCL uses the consolidated signals to compute control commands that correspond to the pilot demand and supress impact of atmospheric disturbance on flight dynamics. It comprises several sub-functions that can be assigned to:

- Command & Stability Augmentation Functions, or
- Envelope Protection Functions.

While the normal mode FCL operates in a clean well-defined (digital) environment, the IM/C function has a connection to the external world, and therefore is susceptible for physical faults of sensors and input devices (e.g. air data sensor freezing or blockage). Sometimes certain failure modes of the systems providing inputs to the IM/C function are missed in systems design, as the accident of Qantas Flight 72 in 2008 showed. There, a specific ADIRU (Air Data Inertial Reference Unit) failure mode led to an erroneous consolidation of the angle of attack value, and eventually to a spurious activation of the angle of attack protection function in the FCL [13].

The IM/C function guarantees signal integrity for the normal mode FCL. It is an independent system function that does not belong to the flight control laws that shall be monitored. Additional monitoring may become necessary to avoid accidents (and incidents) as Qantas Flight 72 [13], LH 1829 [14] or GXL888T [15]. The separation of the IM/C function and the FCL is also reflected in the EASA tender "Horizon Europe Project: Flight Control Laws and Air Data Monitors" (EASA.2021.HVP.28) [16]. That is divided into two lots: the first investigates FCL monitors and the second air data monitors. Both monitors are assumed to be independent. Here, it is assumed that all signals used by the FCL and the Independent Monitor are correct. Signal integrity is assured by the IM/C function and its dedicated monitor.

2.2 Monitoring Level

Figure 2 shows an example of a sequence of events in which a software error leads to a failure condition at aircraft level. The Independent Monitor can work on three levels: software level, system level or aircraft level. Ideally, the Independent Monitor would detect faults on the (FCL) software level, to avoid system failures and eventually hazardous failure conditions. However, the consequences on aircraft level are not easy to predict on this level. As the design objective is to detect failures that lead to hazardous or catastrophic failure conditions, observation of a failure and assessment of its consequences can be better performed on aircraft level.

DOI: 10.13009/EUCASS2023-849

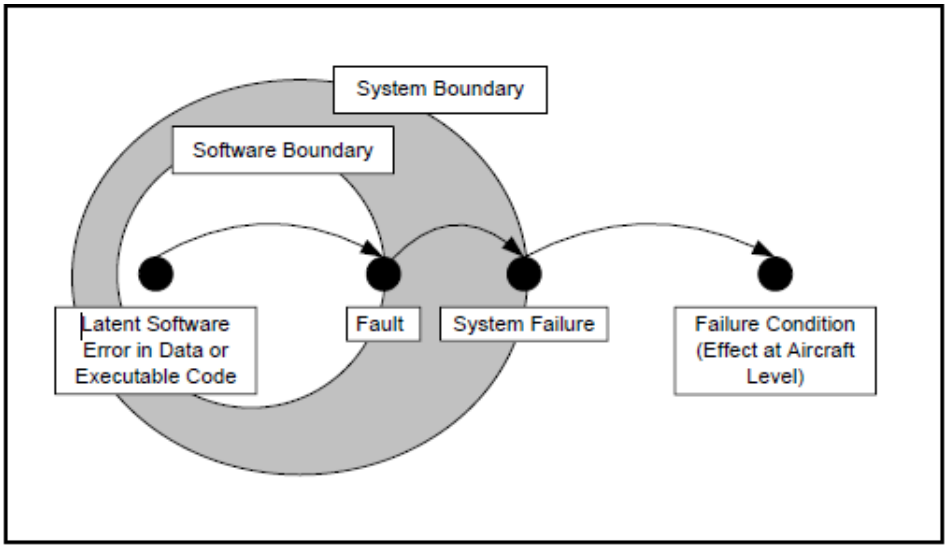

Figure 2: Sequence of Events for Software Error Leading to a Failure Condition (source: [17]).

The block diagram in Figure 3 shows a simplified pilot aircraft control loop and three possible options for FCL monitors (green, blue and orange). The FCL monitoring function can compare different information from different sources. The green monitor compares pilot control inputs with FCL outputs, the blue monitor with control surface positions and the orange monitor with the aircraft reaction.

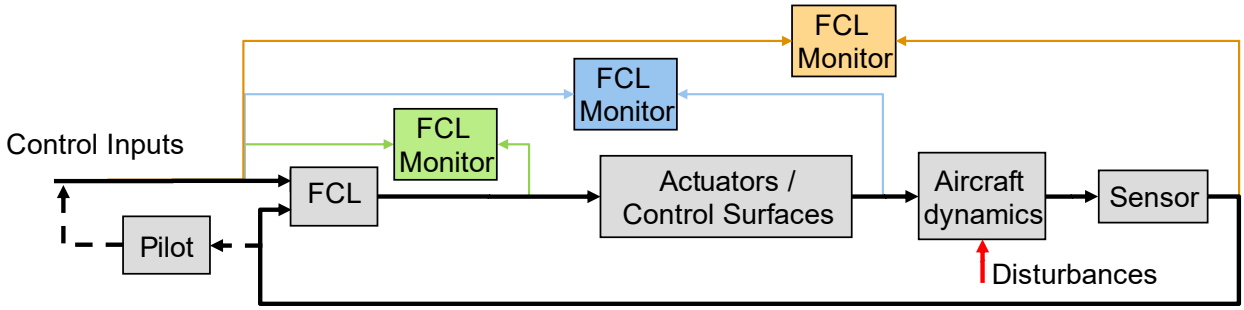

Figure 3: Options for an Independent Monitor

The **green FCL monitor** compares the pilot demand and the FCL output and checks for plausibility. The source for possible faults can be directly isolated to the FCL software. However, it is challenging to achieve functional independence between the monitor and the FCL, as it works on the same level as the item (normal mode FCL) to be monitored, and to assess (and extrapolate) the effect of an erroneous FCL output and thus to determine correct monitor thresholds.

The **blue FCL monitor** compares the control inputs to the control surface deflections and checks for acceptability. An advantage of this option – compared to the green FCL monitor – is that it checks the actual control surface position that controls the aircraft. However, it is challenging that possible actuator failures and dynamics shall not result in spurious FCL monitor tripping. If the actuator is assumed to be failure free, the control surface position and the FCL output are very similar. Therefore, the same challenges as for the green monitor apply to the blue monitor. Additionally, potential actuator dynamics and failures must be considered to avoid false detections.

The **orange FCL monitor** compares the pilot demand to the aircraft reaction. Monitoring of aircraft parameters allows a direct assessment of the criticality of potential failure conditions. Achievement of functional independence seems to be feasible on this level. As aircraft response to external disturbances, such as wake vortex encounters or severe gusts, can also be significant, it is challenging to unambiguously distinguish between a potential FCS malfunction and external disturbances.

In summary, the Independent Monitor can monitor on aircraft, FCS (actuator position) or (FCL) software level. A monitoring on FCS level does not seem to offer significant advantages. Monitoring on the software level has the advantage of a direct allocation of the detected fault and a potentially earlier detection, that would give the pilot more

DOI: 10.13009/EUCASS2023-849

time to avoid or recover from a potentially hazardous failure condition. Functional independence and a direct assessment of the criticality of the failure only seem to be feasible on the aircraft monitoring level. Therefore, a monitor similar to the orange FCL monitor is preferred. In addition, the FCL output may be monitored for fault localisation and isolation and to detect faults before critical aircraft conditions are reached.

2.3 Detection Measures

The third question addresses potential fault detection measures that can be applied to detect FCS failures. Anderson and Lee have proposed a classification of fault detection measures that can be provided in a computer system. Usually, the detection measures work on a local level (within the software or at its output) and are based on the software design or specification [8]. Therefore, they are not suited for the Independent Monitor. However, two fault detection measures can be applied: replication checks and reasonableness checks.

Replication Checks compare the results of redundant components or systems. A fault is detected when the outputs of the variants differ [8], [18]. This check works on FCL level. Dissimilarity in the high-level requirements of the FCL function (functional independence) is necessary, to avoid common mode errors.

Reasonableness Checks are based on a knowledge of the internal design and construction of a system. These checks verify whether the behaviour of the software is acceptable rather than correct, based on predictions on the anticipated system state [8], [18]. Predictions must be derived from aircraft and/or system requirements. They cannot be derived from high-level FCL requirements. In this way, errors common to the Independent Monitor and to the FCL can be avoided.

Potential Independent Monitors can be classified by fault detection measure.

3. Concepts for an Independent Monitor

Concepts for Independent FCL Monitors can be categorized by their decision mechanism. A decision mechanism is a function that adjudicates, arbitrates, or otherwise decides on the acceptability of the results obtained by the FCL variants. Two basic concepts are investigated:

- *Comparator*, and
- *Acceptability Check*.

3.1 Comparator

Comparators are based on the idea of replication checks and work on FCL level. They compare the outputs of the normal mode FCL to the outputs of a functional independent alternative FCL. All safety critical outputs of the normal mode FCL need to be monitored, i.e. compared. However, the alternative FCL may not generate all safety critical outputs. Therefore, it has to be considered that not all normal mode FCL outputs can be compared. In this case a separate monitoring of the missing FCL outputs is necessary. As the functionality of the dissimilar alternative may significantly differ from normal mode FCL, the outputs may significantly differ as well. A comparator that can tolerate such differences is required.

Functional dissimilar alternatives can be an existing backup law (e.g. direct mode FCL), or a newly designed FCL. A newly developed alternative FCL entails a significant additional effort, increases complexity and the risk to implement new errors. Therefore, the development of a new FCL is not further considered here to mitigate the effects of FCL development errors.

All fly-by-wire aircraft have a backup law, that can be used as an alternative. Advantages are, that no extra resources are required for its development and that the backup law is functionally independent from the normal mode FCL. So, it is worthwhile to investigate how the existing alternatives can be used for fault detection. However, operations near the flight envelope limits where mode transition occur, and protection functions are activated, that do not exist in the backup law, may be challenging. Also, some safety critical FCL outputs, e.g. trimmable horizontal stabilizer command, are not generated by the backup law. Additional monitoring of the missing FCL outputs is required.

Simulations of an exemplary CS-25 fly-by-wire aircraft have shown, that a comparison of the normal mode FCL and the backup FCL outputs is possible. In the investigated testcases, the normal mode and the backup FCL simultaneously

DOI: 10.13009/EUCASS2023-849

compute their commands, but only the normal mode FCL controls the aircraft. To achieve acceptable monitoring thresholds, the backup FCL commands were adjusted to the dynamic pressure, see equation (1).

$$\eta_{cmd,BU\ adj} = \frac{\bar{q}_{ref}}{\bar{q}} \eta_{cmd,BU} \quad \textbf{(1)}$$

Figure 4 shows the simulation results for sinusoidal side-stick pitch inputs at cruise condition. The top timeline displays the side-stick pitch inputs (solid line) and vertical wind component (dashed line) over time. The middle timeline shows the elevator commands of the normal mode FCL (solid line) and the backup FCL (dashed line). At the bottom the difference between FCL elevator commands $\Delta\eta_{cmd} = \eta_{cmd,NM} - \eta_{cmd,BU}$ (dashed line) and difference with adjusted backup FCL command (solid line) are displayed.

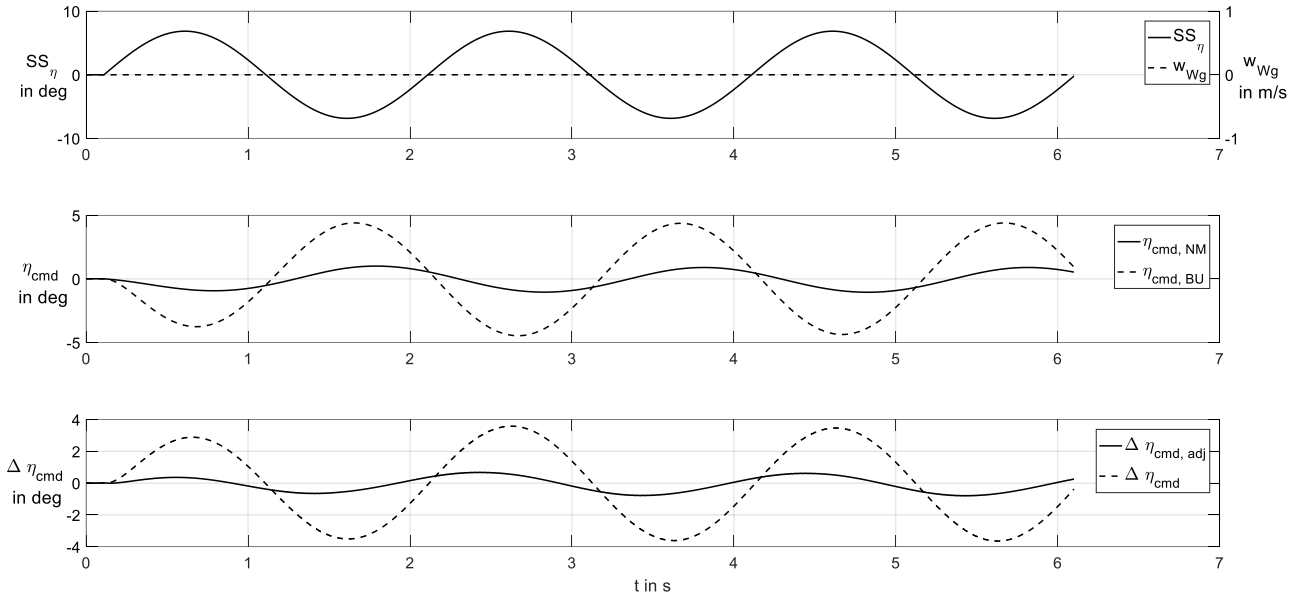

Figure 4: Comparison of FCL outputs for sinusoidal pitch inputs.

It is clearly visible that an adjustment of the backup FCL output with the dynamic pressure can significantly reduce the difference of compared FCL outputs.

Figure 5 shows the simulation results for a discrete downwind gust at cruise condition. The discrete gust, with a probability of occurrence of $\frac{1}{17000} fh \approx 1.43 \cdot 10^{-5} fh$, is designed according to CS 25.341 [3].

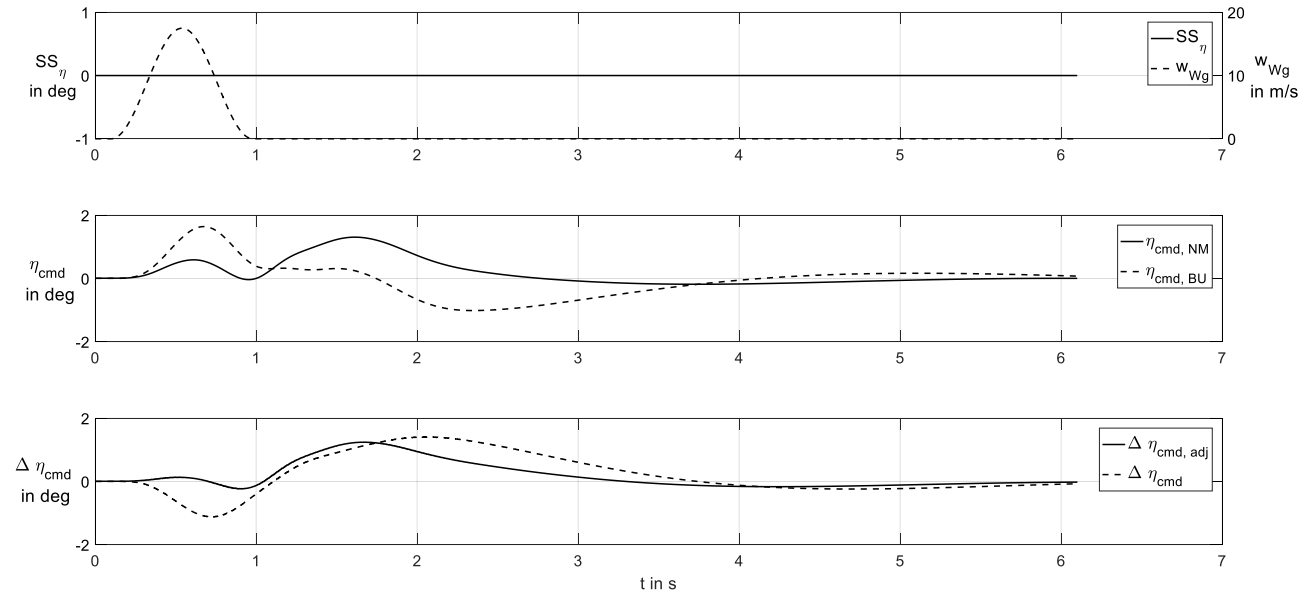

Figure 5: Comparison of FCL outputs for a discrete downwind gust.

DOI: 10.13009/EUCASS2023-849

In both cases the FCL outputs are similar and comparator thresholds, resulting from the difference between the normal mode and backup FCL, are smaller than two degrees surface deflection. This may be acceptable to detect faults that can lead to hazardous or catastrophic conditions if thresholds are exceeded.
Figure 6 schematically illustrates the Independent FCL Monitor based on the comparator concept. It is composed of a pre-processing function that identifies the flight phase and system condition and calculates tolerance thresholds. Additionally, it adjusts the backup FCL commands to the dynamic pressure. The comparator block compares the normal mode FCL commands to the adjusted backup FCL commands. The output of the Independent Monitor is a discrete signal, which indicates that a fault has been detected.

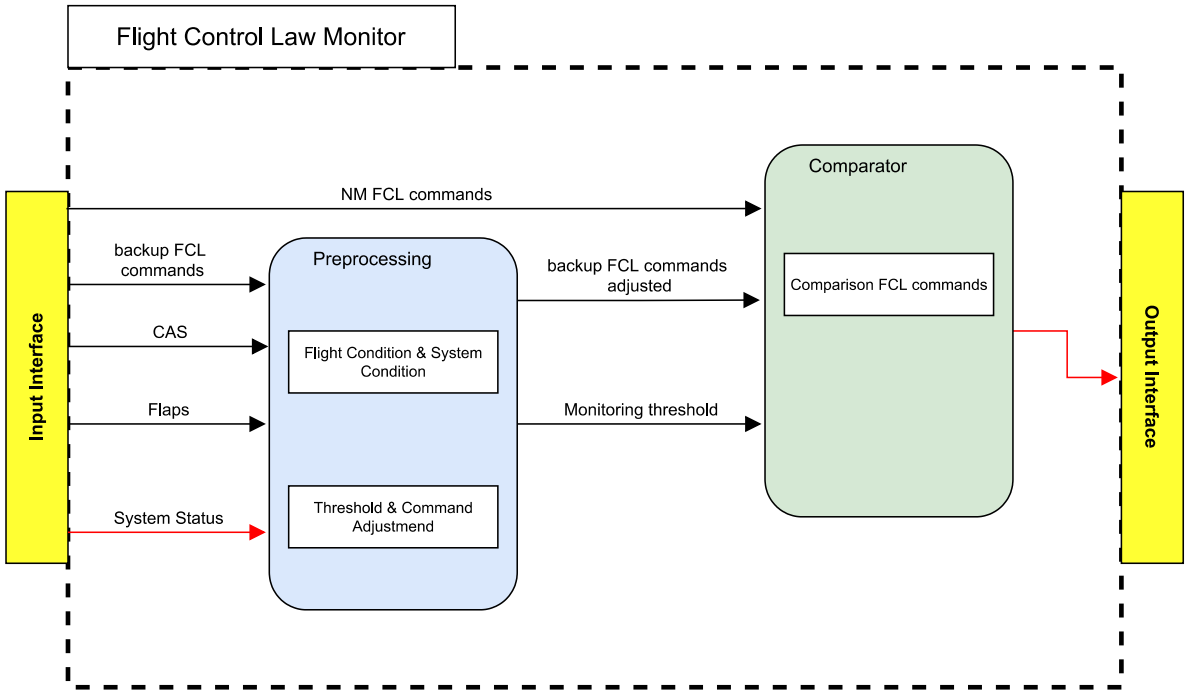

Figure 6: FCL Monitor concept: Comparator.

The required inputs depend on the monitoring tolerances and the required adjustment. The influence of external disturbances and operation at flight envelope limits (e.g., active protections) has to be considered when designing monitoring thresholds. The number of required inputs may be reduced by larger monitoring thresholds. However, it has to be investigated if the resulting granularity of an Independent Monitor with fewer inputs is acceptable for the given monitoring task.

3.2 Acceptability Check

The concept *Acceptability Check* verifies that the behaviour of the FCL software is acceptable and plausible rather than correct, based on predictions on the anticipated system state. Predictions can be derived from aircraft and/or system requirements. Alternatively, they can be derived from flight operations, similar to an instructor pilot observing a student pilot. The instructor knows which inputs are correct and how the aircraft should react on those commands. In this case, the normal mode FCL represents the student pilot, and the Acceptability Check represents the instructor. The acceptability check monitor can work on aircraft and FCL level. Possible Acceptability Checks can be categorized into three groups, which could be used as interacting elements within the acceptability check monitor:

- *Limit Checks*,
- *Behaviour Checks*, or
- *Command Checks*.

Limit Checks check for a violation of (hard) flight envelope limits that the aircraft must not exceed. They are always a sufficient condition to detect failures. If one (or more) of the limits are exceeded, it is assumed that a failure is present. Those limits are derived from aircraft safety requirements, e.g. α_{max} to avoid stall or $n_{z.max}$ to avoid structural damage. Limit Checks are simple and only compare one aircraft state parameter to its respective limit. Table 1 lists possible examples of limit checks.

DOI: 10.13009/EUCASS2023-849

Table 1: Examples of Limit Checks.

Limit	Failure when…
α_{max}	$\alpha > \alpha_{max}$
$V_{CAS,max}$	$V_{CAS} > V_{CAS,max}$
$n_{z,max}$	$n_z > n_{z,max}$

This is similar to the Abnormal Attitude Monitor in AIRBUS aircraft. The Abnormal Attitude Monitor monitors essential flight parameters (static thresholds for pitch attitude, bank angle, angle of attack, calibrated airspeed, and Mach number) [6], [19].

Behaviour Checks check the plausibility of the aircraft reaction under consideration of the pilot demand. These checks alone are never a sufficient condition to determine if the FCL is faulty. If an undesired aircraft response is detected, the cause has to be determined. For example, if the measured normal load factor is significantly greater than one but no pilot command exists, an unexpected aircraft reaction is detected. It can be the reaction to an external disturbance or to system failures (such as an elevator actuator runaway) – or it is caused by an FCL error. If the FCL has output a pitch-up command ($\Delta\eta_{cmd} < 0°$)[2] a failure of the FCS caused by a FCL error is probable. Table 2 gives examples of behaviour checks.

Table 2: Examples of Behaviour Checks.

Condition	Failure detected when…	Rationale
Pilot demands pitch-up AND no protection is active.	($n_z \leq 1$ OR $q \leq 0\,^{\circ}/_{s}$) AND no external disturbance	When pilot demands a pitch-up movement and no protection function is active, the aircraft should pitch up (positive pitch rate q) and build up a positive normal load factor.
Pilot demands right roll AND no protection is active.	$p \leq 0\,^{\circ}/_{s}$ AND no external disturbance	When pilot demands a right roll rate p and no protection function is active, the aircraft should roll to the right $(p > 0\,^{\circ}/_{s})$.

Command Checks comprise checks for acceptability of the FCL commands to the control surfaces that are monitored under consideration of the pilot demand. Predictions on the FCL commands can be derived from aircraft and/or system requirements, but never from FCL requirements. Command Checks have to consider more than one aircraft state parameter to detect failures. Therefore, they are more complex than Limit Checks. Checks of this category can be a sufficient or a necessary condition to detect faults of the FCL. Table 3 gives examples for command checks.

[2] Note that the axis system defined in [20] is used.

DOI: 10.13009/EUCASS2023-849

Table 3: Examples of Command Checks.

Condition	Failure detected when….	Rationale
Aircraft at stall protection limit AND no pitch-up demand.	$\Delta\eta_{cmd} < 0°$	When aircraft is near the stall protection limit and pilot does not demand pitch up, the FCL should command a pitch down to decrease α.
Abnormal pitch-down detected (pilot demands pitch-up). AND no protection is active.	$\Delta\eta_{cmd} > 0°$	When pilot demands a pitch-up movement, but the aircraft does not react as expected because of an FCL pitch-down command. A failure has occurred.

Figure 7 shows the schematic of an Independent FCL Monitor based on the concept of acceptability checks. It is composed of a pre-processing function that may identify the flight phase, system condition and possibly calculate thresholds, and the acceptability check function that includes all plausibility checks to verify that the normal mode FCL commands are acceptable.

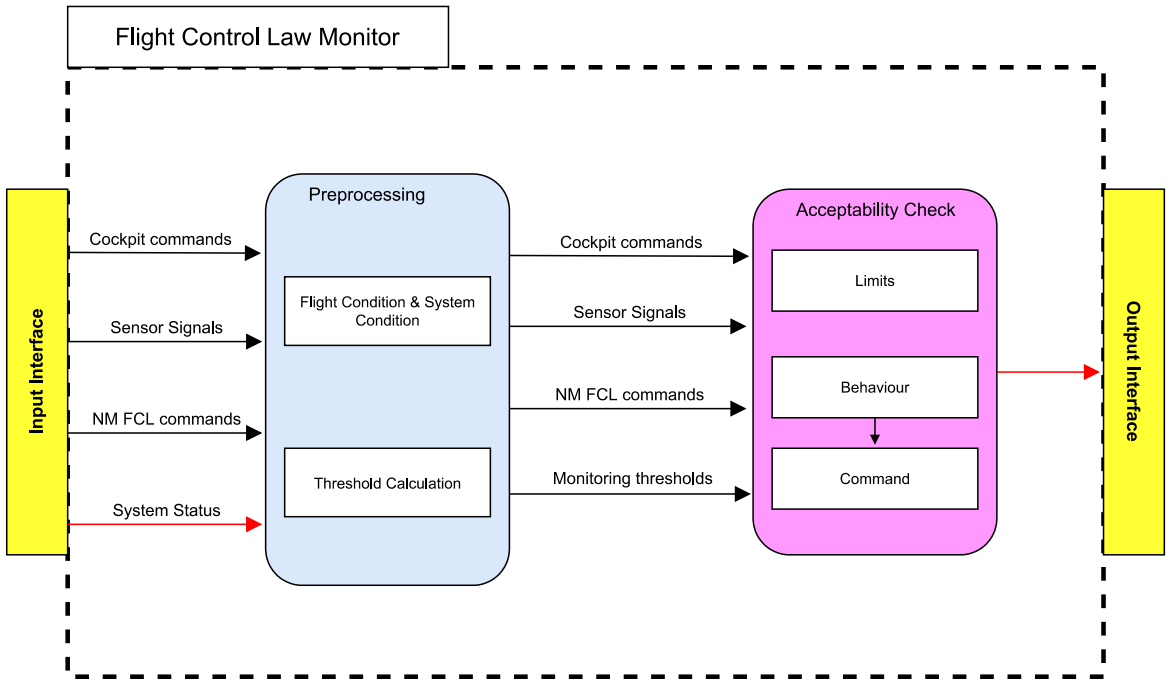

Figure 7: FCL Monitor concept: Acceptability Check.

The output of the Independent FCL Monitor is a discrete signal indicating that a failure has been detected. The required inputs depend on the checks performed by the monitor and should be as few as possible.

4. Conclusions

Electronic Flight Control Systems and the embedded FCL software are highly complex and safety critical. Development assurance alone is not necessarily sufficient to establish an acceptable level of safety. Therefore, architectural means, i.e. fault tolerance, are applied to meet the safety objectives. However, FCL development errors can be the source for common mode errors and failures. A possible solution is an Independent Monitor of the FCL. In this paper principles and concepts for such a monitor are discussed, and two promising concepts are proposed: Comparator and Acceptability Check.

DOI: 10.13009/EUCASS2023-849

The Comparator verifies the correct functionality of the FCL, by comparing the outputs of two functionally independent FCL. It monitors on FCL level and therefore can detect faults before they lead to hazardous failure conditions. The concept Acceptability Check uses plausibility checks to verify that the FCL outputs are acceptable rather than correct. It works on aircraft and FCL level. However, complex checks may be required to monitor on the FCL level. First investigations showed that both concepts are promising.

The division of the concepts is merely academic. In reality an Independent FCL Monitor might by a hybrid of both concepts, combining the early detection potential of the comparator with the global applicability of the acceptability checks.

The next step is to set up a representative closed-loop simulation environment with failure-injection capabilities, to validate the described concepts regarding its effectiveness (correct failure detection) and robustness (no spurious alarms). Different flight phases, flight manoeuvres, flight conditions (including gusts and turbulence) and failure conditions will be evaluated.

Acknowledgements

The work presented in this paper was funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK) as part of its Federal Aviation Research Programme (LUFO VI-1) on the basis of a decision by the German Bundestag within the scope of the joint research project MODULAR and TU Berlin's partner project MODULAR-TUB (grant number 20Y1910C). The authors gratefully acknowledge the support.

References

[1] CAST. 2006. Reliance on Development Assurance Alone When Performing a Complex and Full-Time Critical Function. Position Paper CAST 24. Certification Authorities Software Team.

[2] Rierson, L. 2013. Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance. CRC Press. Boca Raton.

[3] EASA. 2020. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes. CS-25 Amendment 26. European Union Aviation Safety Agency.

[4] SAE Aerospace. 2010. Aerospace Recommended Practice: Guidelines for Development of Civil Aircraft and Systems. SAE ARP4754A. Society of Automotive Engineers.

[5] Traverse P. 2015. Airbus Electrical Flight Controls: A Family of Fault-Tolerant Systems. In: *Digital Avionics Handbook, C. R. Spitzer, U. Ferrel and T. Ferrel. 3rd edition*. CRC Press. Boca Raton.

[6] Traverse, P., I. Lacaze, and J. Souyris. 2006. Airbus Fly-by-Wire: A Process Toward Total Dependability. In: *25th International Congress of the Aeronautical Sciences*.

[7] Flühr, H. 2012. Avionik und Flugsicherungstechnik. 2. Auflage. Springer Verlag. Heidelberg.

[8] Torres-Pomales, W. 2000. Software Fault Tolerance: A Tutorial. NASA/TM-2000-210616. National Aeronautics and Space Administration.

[9] Yeh, y. C. 1996. Triple-triple redundant 777 primary flight computer. In: *1996 IEEE Aerospace Applications Conference*. 293-307 vol. 1.

[10] Leveson N. G. 2011. Engineering a Safer World: System Thinking Applied to Safety. MIT Press. Cambridge, MA.

[11] EASA. Certification Review Item: Consideration of Common Mode Failures and Errors in Flight Control Functions. Generic CRI D-XXX. European Union Aviation Safety Agency.

[12] Fielding, C. and R. Luckner. 2000. Industrial Considerations for Flight Control. In: *Flight Control Systems: practical issues in design and implementation, R. W. Pratt.* The Institution of Electrical Engineers. Cornwall.

[13] ATSB. 2011. Final: In-flight upset 154 km west of Learmonth, WA 7 October 2008 VH-QPA Airbus A330-303. AO-2008-070. Australian Transport Safety Bureau.

[14] BFU. 2015. Interim Report. BFU 6X014-14. German Federal Bureau of Aircraft Accident Investigation.

[15] BEA. 2010. Report: Accident on 27 November 2008 off the coast of Canet-Plage (66) to the Airbus A320-232 registered D-AXLA operated by XL Airways Germany. BEA d-la081127. Bureau d'Enquêtes et d'Analyses pour la sécurité de l'aviation civile.

[16] EASA. 2021. Horizon Europe Project: Flight Control Laws and Air Data Monitors. EASA.2021.HVP.28. Online. Accessed 16.05.2023. https://etendering.ted.europa.eu/cft/cft-display.html?cftId=9764.

[17] RTCA. 2011. Software Considerations in Airborne Systems and Equipment Certification. RTCA DO-178C. Radio Technical Commission for Aeronautics.

DOI: 10.13009/EUCASS2023-849

[18] Lee, P. A. and T. Anderson. 1990. Fault Tolerance Principles and Practice. 2nd edition. Springer Verlag. Wien.

[19] Favre, C. 1994. Fly-by-Wire for commercial aircraft: the Airbus experience. In: *International Journal of Control*. Vol. 59, No. 1, pp. 139-157.

[20] Brockhaus, R., W. Alles, and R. Luckner. 2011. Flugregelung. 3. Auflage. Springer Verlag. Heidelberg.