

METHODOLOGY AND V1 PROTOTYPE OF AN ADAPTIVE/ADAPTABLE MULTI-UAV GCS HMI IN A FLOOD SAR CO-SIMULATION TESTBED

Andrew Sarmento[1], Petter Krus[2] & Emília Villani[1]

[1]Mechanical Engineering Division, Aeronautics Institute of Technology, Praça Marechal Eduardo Gomes, 50 - Vila das Acácias, 12228-900, São José dos Campos, SP, Brazil
[2]Department of Management and Engineering, Linköping University (LiU), Campus Valla, 581-83 Linköping, Sweden

Abstract

Multi-UAV operations are increasingly moving toward supervisory-control paradigms in which a single operator monitors and commands multiple vehicles. In time-critical missions such as search-and-rescue (SAR), abnormal events such as propulsion degradation, control-surface malfunction, sensor degradation, or communication loss may rapidly increase operator workload and decision complexity. This paper presents a structured methodology and a first implemented version (V1) of an adaptive/adaptable Human-Machine Interface (HMI) for a multi-UAV Ground Control Station (GCS). The proposed methodology links operational scenario definition, task analysis, function decomposition, Levels of Autonomy (LOA), adaptation-policy design, transparency mechanisms, and evaluation variables. The methodology is instantiated in a co-simulation testbed that integrates MATLAB/Simulink R2024b, JSBSim-based fixed-wing flight dynamics, Unreal Engine 5.3, Cesium ion, UDP communication, and synchronized logging. The V1 prototype demonstrates the feasibility of supervising three fixed-wing UAVs in a georeferenced flood-response SAR scenario, including multi-UAV visualization, sensor-camera views with telemetry overlays, and map-based command-and-control tools for waypoint, orbit, flight-mode, speed, altitude, and basic vehicle commands. The results reported in this paper are implementation and integration results rather than formal human-subject validation. The prototype establishes the infrastructure required for future experiments comparing human-operated, human-delegated, and human-supervised LOA configurations under nominal and degraded mission conditions.

Keywords: Multi-UAV supervision, Ground Control Station (GCS), Adaptive/Adaptable Human–Machine Interface (HMI), Co-simulation testbed

1. Introduction

Multi-UAV operations are moving toward supervisory control paradigms in which a single operator monitors and commands multiple vehicles, particularly in time-critical missions such as search-and-rescue (SAR). Although increased autonomy can reduce operator workload, real missions include dynamic conditions and abnormal events, such as degraded propulsion, control-surface malfunctions, sensor issues, and communication loss, that may rapidly increase cognitive demand and decision complexity. Human factors analyses of UAV incidents highlight the importance of interface design, procedures, and automation management under operational stressors [1]. Consequently, future Ground Control Stations (GCS) and their Human–Machine Interfaces (HMI) must go beyond static layouts and support both *adaptable* (operator-driven) and *adaptive* (system-driven) changes in information presentation and autonomy allocation, typically framed through Levels of Automation (LOA) [2]. Situation awareness considerations further motivate design choices that preserve operator comprehension during high automation and transitions [3]. Recent literature on human–autonomy teaming emphasises the need for experimental platforms that enable rigorous, comparable evaluation of interface and autonomy strategies [4].

A practical challenge is the lack of end-to-end, repeatable, and sufficiently realistic evaluation platforms that integrate: (i) multi-UAV supervision, (ii) mission-phase-dependent automation, (iii) systematic fault injection, and (iv) logging/metrics that enable fair comparisons between autonomy and interface strategies [5, 4]. This research addresses this gap by presenting (1) a structured methodology for developing an adaptive/adaptable multi-UAV GCS HMI and (2) a first implemented version (V1) integrated into a high-fidelity co-simulation testbed for a flood-response SAR context. The testbed couples MATLAB/Simulink (2024b) [6] and JSBSim-based flight dynamics [7] with an Unreal Engine 5.3 visualisation [8] and a georeferenced environment using Cesium ion [9], exchanging telemetry and commands via User Datagram Protocol (UDP). The scope of this paper is the *methodology* and the *V1 prototype demonstration*. Formal human-subject validation and physiological workload analysis are planned as follow-on work [10] and are not reported here.

1.1 Objectives

The objective of this work is to define and demonstrate a repeatable development methodology for adaptive/adaptable HMIs applied to single-operator, multi-UAV missions. The methodology is instantiated in a runnable V1 prototype for the supervision of three fixed-wing UAVs in a flood SAR scenario, with emphasis on LOA, operator interaction, mission monitoring, and decision-support functions.
The main contributions of this paper are threefold. First, it proposes a traceable methodology for the development of adaptive/adaptable multi-UAV HMIs, covering scenario definition, task analysis, LOA specification, adaptation-policy design, and implementation guidelines for transparency, operator authority, and manual override mechanisms [2, 3]. Second, it presents a co-simulation testbed architecture that enables real-time execution of a multi-UAV mission with synchronised logging of vehicle states, mission events, and operator interactions, integrating flight-dynamics simulation, decision-making logic, and a photorealistic visualisation environment [4, 6, 7, 8, 9]. Third, it demonstrates a V1 HMI prototype that supports multi-UAV monitoring through camera and sensor views, mission overlays, and command-and-control tools, including waypoint assignment, orbit commands, flight-mode selection, and basic vehicle actions.

2. HMI Design Methodology

The proposed HMI design methodology defines a traceable process for developing adaptive/adaptable interfaces for single-operator, multi-UAV missions. The methodology links the operational scenario, mission phases, operator tasks, automation functions, LOA, interface elements, and evaluation variables. In this way, the HMI is specified not only as a graphical display, but as the interaction layer that connects the operator, the UAV fleet, the mission-management logic, and the autonomy-management policy.
Figure 1 summarises the methodology applied to a multi-UAV Ground Control Station (GCS). The process is organised into seven design steps. Each step generates explicit artefacts that can be implemented in the co-simulation architecture and later evaluated through human-in-the-loop experiments.

2.1 Operational Scenario and Task Analysis

The first step defines the reference mission, its phases, the operator tasks, and the relevant contingency events. In this work, the reference scenario is a flood Search-and-Rescue (SAR) mission in which a single operator supervises three fixed-wing UAVs. The mission includes transit to the area of interest, area search, target confirmation, possible re-planning after failures, and return to base.
For each mission phase, the task analysis identifies the information required by the operator, the decisions expected from the operator or the automation, and the safety-critical events that may require intervention. At the fleet level, the operator must monitor global mission progress, area coverage, task allocation, remaining mission time, and the status of re-planning actions. At the vehicle level, the operator must monitor position, altitude, speed, flight mode, autonomy margin, sensor status, communication status, and active warnings.
This step follows situation-awareness-oriented design principles, in which the interface must support the operator's perception of relevant elements, comprehension of their meaning, and projection of

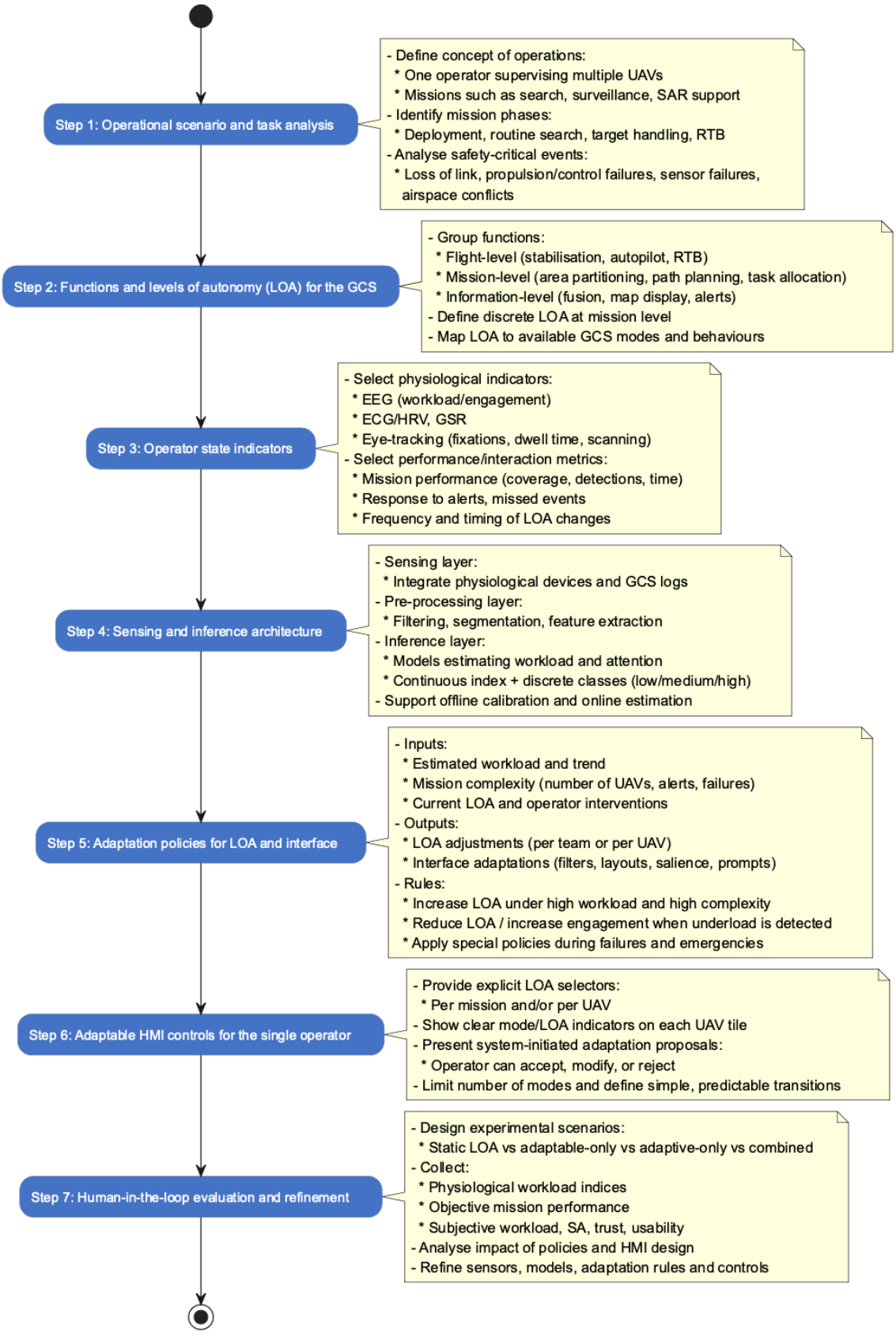


Figure 1 – Application of Adaptive/Adaptable HMI to a Multi-UAV GCS.

their future state [3]. The resulting artefacts are the mission-phase model, the operator-task list, the information-requirement matrix, and the list of relevant contingency events.

2.2 Function Decomposition and LOA Specification

The second step decomposes the GCS functions according to the role assigned to the human operator and to the automation. Following established LOA models, functions are considered in terms of information acquisition, information analysis, decision selection, and action implementation [2]. For the proposed multi-UAV GCS, these functions are grouped into three categories:

1. **Flight-level functions:** stabilisation, waypoint following, altitude and speed control, automatic take-off, automatic landing, and emergency return-to-base;
2. **Mission-level functions:** area partitioning, path generation, task allocation, task reallocation after failures, and selection of candidate re-planning strategies;
3. **Information-level functions:** telemetry visualisation, camera and sensor views, map overlays, alert generation, alert filtering, and presentation of re-planning options.

Flight-level functions remain mostly automated for safety and workload reasons. Mission-level functions are the main focus of LOA variation because they determine how planning, task allocation, and failure response are shared between the operator and the system. Information-level functions support the operator by defining what information is displayed, highlighted, filtered, or prioritised according to the mission phase, UAV state, and event condition.
The methodology considers three representative LOA configurations for mission management. In the *human-operated* configuration, the operator directly selects mission actions and responds to failures, while low-level flight functions remain automated. In the *human-delegated* configuration, the system proposes candidate actions or re-planning alternatives, and the operator selects the action to be executed. In the *human-supervised* configuration, the system selects the preferred action according to the mission logic, while the operator monitors the decision and retains the authority to modify, reorder, or cancel it [2, 4].
The output of this step is a function LOA allocation matrix that defines which functions are manual, automated, delegated, or supervised in each mission phase and event condition.

2.3 Operator-State, Performance, and Interaction Indicators

The third step defines the indicators required to evaluate the HMI and to support future adaptive behaviour. The methodology considers three groups of indicators:

- **Mission-performance indicators:** total mission time, time per mission phase, area covered, number of inspected targets, missed detections, re-planning success, response time to warnings, and successful return of the UAVs;
- **Interaction indicators:** number of commands, LOA changes, waypoint modifications, accepted or rejected recommendations, orbit commands, flight-mode changes, and manual overrides;
- **Operator-state indicators:** optional physiological and behavioural measures, including EEG, ECG/HRV, GSR, and eye-tracking features, planned for later stages of the research [10, 11, 12].

These indicators connect the interface design with later performance, workload, and situation-awareness evaluation. In this methodology, operator-state indicators are treated both as evaluation variables and as possible inputs for future adaptive behaviour. The output of this step is a measurement and logging plan specifying what must be recorded, how it is time-synchronised, and how it will be related to mission phase, LOA, failures, and operator actions.

2.4 Sensing, Logging, and Inference Architecture

The fourth step defines the data architecture required to connect telemetry, mission events, operator interactions, and optional physiological data. This architecture is organised into three layers.

The *sensing and logging layer* records UAV states, camera or sensor information, mission events, warnings, LOA changes, and operator commands. The *preprocessing layer* organises these data into time-aligned records indexed by UAV, mission phase, LOA, and event condition. When physiological data are included, this layer also processes the raw signals and extracts features such as EEG band powers, heart-rate variability indices, GSR statistics, and eye-tracking measures [13, 14, 10]. The *inference layer* estimates higher-level operator-state variables, such as workload or attention state, and provides these estimates to the adaptation policy.

For implementations that do not yet include closed-loop adaptation, the inference layer may initially be used only for offline analysis. However, the methodology requires that the architecture support time-synchronised data recording, so that adaptive behaviour can be incorporated in later development stages without redesigning the data pipeline.

2.5 Adaptation Policy Design

The fifth step defines how mission context, detected events, and operator-related indicators are mapped to LOA or interface changes. The adaptation policy receives three main input groups: mission phase, event or failure state, and operator-related indicators. Its outputs may include LOA recommendations, interface highlighting, alert prioritisation, re-planning options, or changes in the level of automation assigned to specific functions.

The policy can be implemented in two complementary forms. In an *adaptable* implementation, the operator explicitly selects the LOA, command mode, or interface configuration. In an *adaptive* implementation, the system recommends or triggers changes based on mission state, failure condition, interaction history, or estimated workload. The present methodology is defined to support both cases, allowing manual, semi-manual, and adaptive autonomy strategies to be compared within the same experimental framework [15, 16, 17, 18].

Each adaptation rule must define its trigger condition, expected action, authority level, and feedback to the operator. This requirement is intended to keep adaptive behaviours predictable, explainable, and compatible with operator authority. For safety-critical actions, the rule must also specify whether confirmation is required, whether the operator can override the action, and which constraints cannot be violated.

2.6 Adaptable Controls and Transparency Mechanisms

The sixth step specifies the interface controls that allow the operator to understand, select, and override automation. Adaptable controls include LOA selectors, flight-mode selectors, waypoint commands, orbit commands, speed and altitude commands, and basic vehicle actions. These controls allow the operator to manage the mission while retaining authority over automation decisions.

Transparency is provided through mode indicators, UAV status panels, warnings, planned-path visualisation, feedback messages, and telemetry overlays. Each UAV must show its current flight mode, mission state, LOA, health status, communication status, and active alerts. When the system proposes or registers an action, the interface must make clear whether it is a recommendation, a pending action requiring confirmation, or an executed command [3, 4, 19].

These mechanisms are intended to reduce ambiguity in mode management and mitigate out-of-the-loop effects during high-automation conditions. They also define the HMI requirements that guide implementation in any prototype derived from the proposed methodology.

2.7 Human-in-the-Loop Evaluation and Refinement

The final step evaluates the HMI through human-in-the-loop experiments. The evaluation compares LOA configurations and mission conditions using performance metrics, interaction logs, subjective workload measures, and optional physiological indicators. The main configurations considered in this methodology are human-operated, human-delegated, and human-supervised modes.

The logged data allow analysis at three levels. At the mission level, total performance can be compared across LOA configurations. At the phase level, the analysis can identify which mission segments produce higher workload or require more interaction. At the event level, the analysis can evaluate how the operator responds to specific failures and how effectively the interface supports decision-making.
The results of this evaluation are used to refine the methodology artefacts: task definitions, information requirements, function LOA allocation, adaptation policies, interface controls, transparency mechanisms, and logging requirements. In this way, the methodology supports iterative development rather than treating the HMI as a fixed design.

2.8 Methodology Outputs

The complete methodology produces a set of artefacts that connect the conceptual HMI design to the co-simulation architecture and to the later experimental analysis. These artefacts include the mission-phase model, operator-task list, information-requirement matrix, function LOA allocation matrix, measurement and logging plan, sensing and inference architecture, adaptation-policy rules, interface-control requirements, transparency requirements, and human-in-the-loop evaluation plan.
Together, these outputs define what must be implemented, displayed, logged, and evaluated in prototype implementations derived from the proposed methodology. The specific implementation results are therefore presented separately in the future section.

3. Co-Simulation Architecture

The co-simulation architecture defines the experimental framework used to implement, connect, and evaluate the adaptive/adaptable HMI methodology in a multi-UAV mission context.
The proposed architecture integrates six functional layers: vehicle dynamics, mission-management logic, visualisation, operator interaction, communication, and data logging. These layers are required to ensure that UAV states, mission events, operator commands, LOA, and evaluation variables are represented consistently during the experiment. Figure 2 summarises the overall information flow.

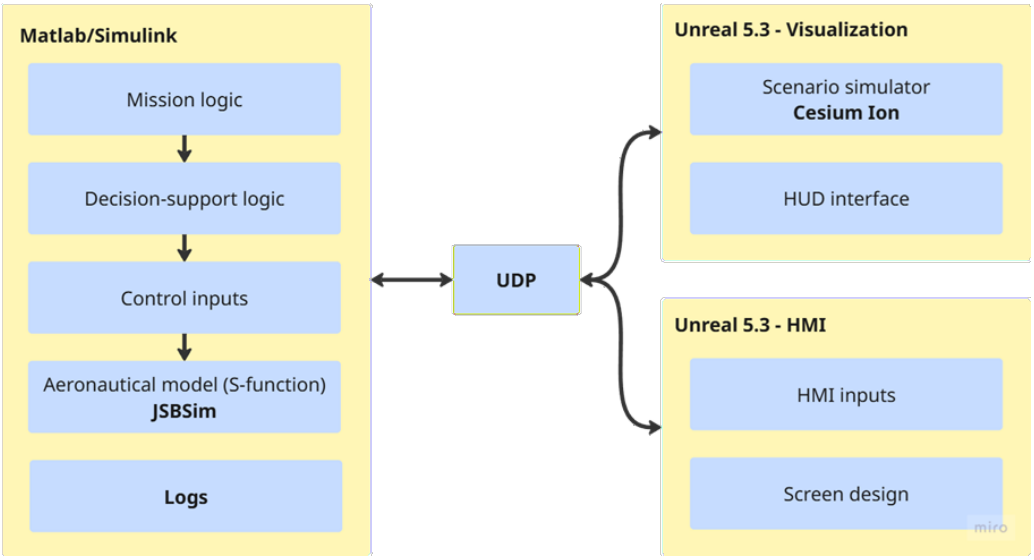


Figure 2 – Overall co-simulation architecture and information flow.

In the proposed framework, the vehicle-dynamics layer provides the simulated flight states of each UAV, including position, attitude, speed, altitude, and variables required for guidance, control, and fault monitoring. The mission-management layer coordinates mission phases, task allocation, replanning logic, failure states, and decision-support functions. The visualisation and interaction layer represents the operational environment, UAV positions, camera views, mission overlays, and HMI elements. The communication layer exchanges telemetry, operator commands, and mission events between the simulation and visualisation environments. Finally, the logging layer records the vari-

ables required for later analysis of mission performance, operator interaction, workload, and LOA transitions.

For the implementation used in this research, the generic layers are mapped to specific software components. MATLAB/Simulink R2024b is used for mission logic, decision-support computation, and integration with the dynamic models [6]. JSBSim-based fixed-wing flight dynamics are used to represent the nonlinear response of each UAV [7]. Unreal Engine 5.3 is used for real-time visualisation and HMI implementation, including a head-up display (HUD) for telemetry and mission-status overlays [8]. Cesium ion provides the georeferenced 3D environment used to represent the flood-response scenario [9]. UDP communication is used to exchange telemetry, commands, and mission events between the simulation and visualisation layers. Synchronized logs are used to record vehicle states, mission states, alerts, LOA changes, and operator actions.

This division of roles separates numerical simulation from visual interaction. The dynamics and mission-management components are responsible for modelling, control logic, event handling, and decision-support computation, whereas the visualisation and HMI components are responsible for presenting the scenario, supporting operator interaction, and displaying the relevant mission information. This separation also makes the framework modular: the flight model, the mission planner, the communication interface, the visual scenario, and the HMI can be modified independently as the research evolves.

3.1 Level-of-Autonomy Representation

The LOA representation defines how autonomy-management logic is connected to the mission context, detected events, and operator-state information. For each UAV u, the active autonomy level at time t is represented as:

$$\mathrm{LOA}_u(t) = \mathscr{G}\big(\phi(t), e_u(t), \hat{w}(t)\big), \tag{1}$$

where $\phi(t)$ denotes the current mission phase, $e_u(t)$ represents the set of detected events or failures affecting UAV u, and $\hat{w}(t)$ is an estimated operator workload state, planned for inclusion in later stages of the system [10]. The mapping $\mathscr{G}(\cdot)$ defines the policy used to select, recommend, or update the appropriate LOA.

In the proposed methodology, $\mathscr{G}(\cdot)$ can be implemented either as an adaptable or adaptive mechanism. In an adaptable implementation, the operator explicitly selects or modifies the active LOA through the HMI. In an adaptive implementation, the system may recommend or trigger LOA changes based on mission phase, failure state, interaction history, or operator-state estimates. In both cases, the HMI must display the active LOA, indicate the reason for any recommendation or transition, and preserve operator authority through confirmation and override mechanisms [2, 4].

The methodology considers three representative LOA configurations for multi-UAV mission management. In the *human-operated* configuration, the operator directly selects mission actions and responds to failures, while low-level flight stabilisation and navigation remain automated. In the *human-delegated* configuration, the system proposes candidate actions or re-planning alternatives, and the operator selects the action to be executed. In the *human-supervised* configuration, the system selects the preferred action according to the mission logic, while the operator monitors the decision and retains the authority to modify, reorder, or cancel it. These configurations provide a common basis for comparing different levels of authority allocation in the same multi-UAV mission scenario.

3.2 Mission Re-Planning and Resource Constraints

The mission-management layer must ensure that any new task assignment or re-planned trajectory remains feasible for the affected UAVs. This requirement is particularly relevant in time-critical SAR missions, where tasks may need to be redistributed after events such as sensor degradation, communication loss, propulsion degradation, or other abnormal conditions. Therefore, candidate plans must be evaluated not only in terms of mission effectiveness, but also in terms of resource feasibility. Mission re-planning feasibility is constrained by the endurance limit of each vehicle:

$$\mathrm{dur}_u^{\mathrm{path}} + \mathrm{dur}_u^{\mathrm{task}} + \mathrm{dur}_u^{\mathrm{loiter}} + \mathrm{dur}_u^{\mathrm{return}} \leq \mathrm{autonomy}(u). \tag{2}$$

In Eq. (2), $\mathrm{dur}_u^{\mathrm{path}}$ is the travel time from the current position of UAV u to the assigned task area, $\mathrm{dur}_u^{\mathrm{task}}$ is the time required to perform the assigned mission task, $\mathrm{dur}_u^{\mathrm{loiter}}$ represents the accumulated time spent in loitering, waiting, turning, or intermediate manoeuvring segments, and $\mathrm{dur}_u^{\mathrm{return}}$ is the time required for UAV u to return from the mission area to the base airfield. Thus, a new assignment is feasible only when the total expected mission duration remains lower than or equal to the available autonomy of the vehicle.
Although Eq. (2) is written in terms of time, the same logic can be extended to range, fuel, or energy limits. In this way, the mission-management layer can reject candidate assignments that would require excessive distance, excessive loitering, or insufficient remaining resources to return safely. These constraints connect the HMI decision-support layer to the physical limitations of the UAV fleet.

3.3 Decision-Support Objectives

The decision-support component ranks or filters candidate plans generated during mission re-planning. Due to the time-critical nature of SAR missions, the re-planning process must consider both mission efficiency and resource constraints under limited operational time [20]. In the proposed methodology, the multi-objective formulation considers three main objectives:

1. **Minimisation of the maximum mission duration among UAVs:** this objective reduces the largest total time assigned to any individual UAV, helping to keep all vehicles within their endurance envelopes.
2. **Minimisation of the time imbalance among UAVs:** this objective distributes the mission workload more evenly across the fleet by reducing the difference between the total mission times assigned to each vehicle.
3. **Minimisation of loitering and turning penalties:** this objective penalises loitering segments and turning manoeuvres, since these manoeuvres increase path length, consume additional time and energy, and are generally inefficient for fixed-wing UAVs.

From the HMI perspective, the objective is not only to compute an efficient plan, but also to present feasible and interpretable alternatives to the operator. In a human-delegated configuration, the system may display multiple candidate plans and their expected mission times, allowing the operator to select the preferred option. In a human-supervised configuration, the system may identify the highest-ranked feasible plan and present it to the operator for monitoring, confirmation, or override. Therefore, the decision-support output must be compatible with the active LOA and with the transparency requirements defined in the HMI methodology.

3.4 Logging and Evaluation Variables

The co-simulation architecture also defines the variables required for later evaluation. For each UAV, the system records telemetry variables such as position, altitude, speed, heading, attitude, flight mode, health state, sensor state, and communication state. At the mission level, it records task assignments, mission phase, active LOA, detected events, re-planning triggers, selected actions, and completion times. At the HMI level, it records operator commands, mode selections, waypoint changes, orbit commands, accepted or rejected suggestions, and manual overrides.
This logging structure supports three levels of analysis. At the mission level, it allows comparison of total mission time, area coverage, number of completed tasks, and success of re-planning. At the phase level, it allows the analysis of workload-sensitive segments such as search, target confirmation, failure management, and return to base. At the event level, it allows reconstruction of the sequence between a detected failure, the HMI alert, the operator response, and the resulting mission action.
The same time base is intended to support future physiological and questionnaire-based evaluation. In later stages, workload indicators derived from electrocardiogram (ECG), galvanic skin response (GSR), electroencephalogram (EEG), eye tracking, Instantaneous Self Assessment (ISA), and NASA Task Load Index (NASA-TLX) can be aligned with mission phases, LOA configurations, and failure events. This will allow the evaluation of whether a given LOA configuration improves mission performance without producing excessive workload or reducing operator situation awareness.

4. Results: V1 Prototype

This section presents the first implemented version of the proposed adaptive/adaptable multi-UAV GCS HMI. The V1 prototype was developed to verify the feasibility of the co-simulation architecture, the georeferenced SAR scenario, the multi-UAV monitoring concept, and the command-and-control workflow. Therefore, the results reported here are implementation and integration results, rather than formal human-subject validation results. The evaluation of workload, situation awareness, and operator performance will be addressed in later experimental stages.

The prototype implements four main capabilities: (i) a georeferenced flood-response scenario in Unreal Engine using Cesium ion; (ii) simultaneous visualisation of three fixed-wing UAVs; (iii) sensor-camera views with telemetry overlays for each UAV; and (iv) a map-based HMI for waypoint management, orbit commands, flight-mode selection, and basic vehicle actions. Table 1 summarises the implemented V1 elements and their role in the overall methodology.

Table 1 – Implemented elements in the V1 prototype.

Implemented element	**V1 capability**	**Purpose in the proposed methodology**
Georeferenced SAR scenario	Flood-response environment in Porto Alegre rendered in Unreal Engine with Cesium ion	Provides a realistic spatial context for mission monitoring, camera interpretation, and SAR task execution.
Multi-UAV visualisation	Simultaneous representation of three fixed-wing UAVs in the same operational environment	Enables the evaluation of single-operator supervision in a multi-vehicle mission context.
Sensor-camera monitoring	Individual camera views with telemetry and vehicle-status overlays	Supports vehicle-level situation awareness and comparison between UAVs during the mission.
Map-based command interface	Waypoint, orbit, speed, altitude, flight-mode, and basic vehicle commands	Provides adaptable controls that allow the operator to retain authority while delegating selected functions to automation.
Mode and status presentation	Visual indication of vehicle states, mission elements, and command options	Supports transparency, mode awareness, and preparation for future LOA-based interaction.
Logging infrastructure	Recording of vehicle states, mission events, interface states, and operator commands	Provides the data foundation for later comparison of LOA configurations, failure responses, and workload-related indicators.

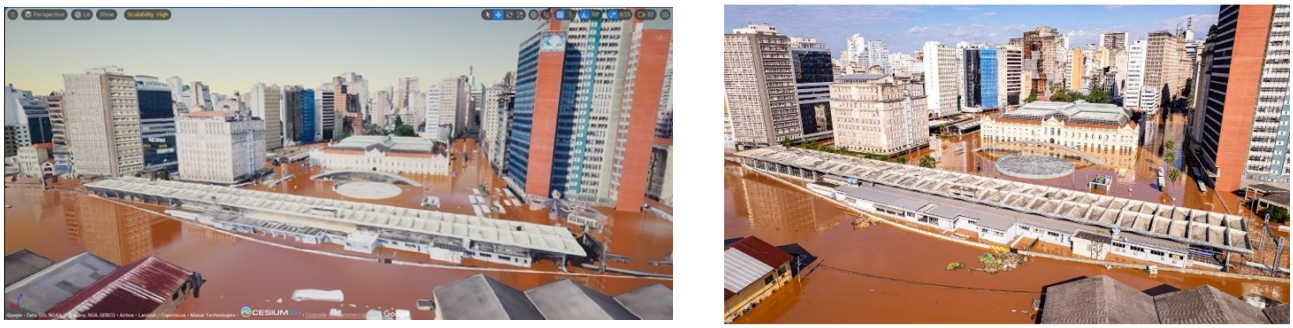
4.1 Scenario Instantiation

The first operational environment was implemented in Unreal Engine using Cesium-ion-based geospatial streaming to represent a flood-response context in Porto Alegre, Brazil. The scenario was designed to provide more than a visual background: it establishes a mission-relevant environment in which the operator can interpret terrain, urban references, UAV trajectories, and sensor viewpoints within a realistic spatial context. This is particularly important for SAR missions, since the operator must relate the position of each UAV and its camera perspective to flooded regions, buildings, possible victim locations, and return routes.

Figure 3 compares the implemented simulation environment with a real image of the flood event. The comparison shows that the V1 scenario reproduces the main spatial cues required for the proposed HMI evaluation, including the urban layout, flooded areas, and elevated viewpoints suitable for UAV camera simulation. These elements are sufficient for the current objective of demonstrating a runnable prototype and preparing the environment for future human-in-the-loop experiments.

4.2 Multi-UAV Monitoring Interface

The V1 prototype supports the simultaneous monitoring of three fixed-wing UAVs. Each vehicle is associated with an individual sensor-camera view, allowing the operator to observe the mission from



(a) **(b)**

Figure 3 – Comparison between the simulated flood-response SAR scenario in Unreal Engine and a real image of the flood event. (a) Flood-response SAR scenario visualisation in Unreal Engine [8, 9]; (b) real image of the flood [21].

the perspective of each UAV. Figure 4 shows the implemented multi-camera interface. Each camera view includes a head-up display (HUD) overlay with relevant telemetry and vehicle-status information, such as speed, altitude, heading, attitude, engine status, and landing-gear status.
This result demonstrates that the visualisation layer can present multiple UAV perspectives in parallel while preserving vehicle-specific information. From an HMI perspective, this is an important requirement for multi-UAV supervision because the operator must maintain awareness of both the global mission and the local state of each vehicle. The camera views support vehicle-level monitoring, while the telemetry overlays reduce the need to search for basic flight information in separate panels.

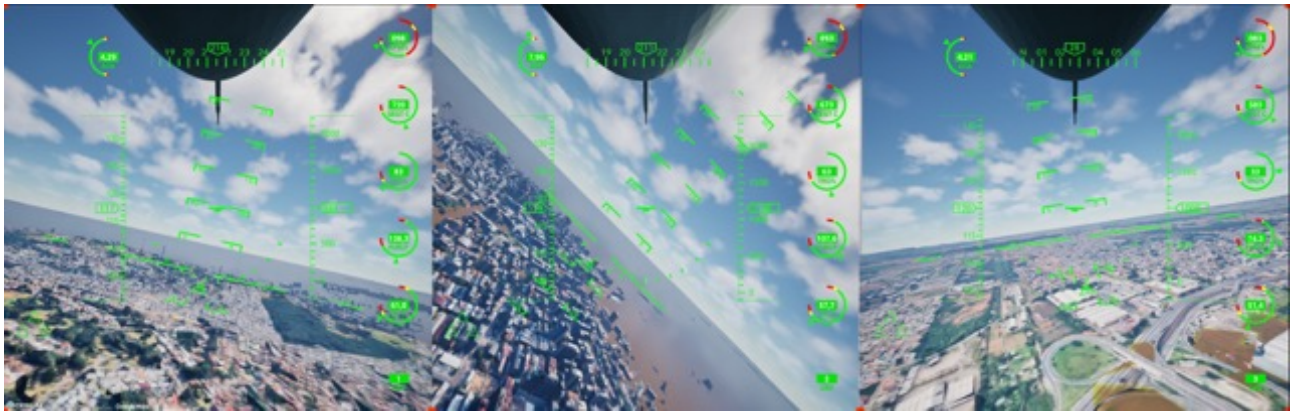


Figure 4 – V1 prototype: multi-UAV camera views with telemetry overlays.

4.3 Command-and-Control Interface

The command-and-control interface was implemented as a map-oriented HMI, as shown in Figure 5. The interface displays UAV positions, planned paths, mission markers, and command widgets over the georeferenced scenario. It allows the operator to issue waypoint commands, define orbit behaviours, monitor UAV position, and select basic vehicle actions.
The implemented actions include brakes, landing-gear commands, engine start/stop, speed commands, altitude commands, and flight-mode selection. The available flight modes include manual control and automated sequences such as automatic take-off, waypoint navigation or cruise, and automatic landing. These functions instantiate the adaptable component of the proposed HMI, since the operator can explicitly select commands and modes while the system maintains automated support for low-level flight functions.
The V1 command interface also provides the first implementation basis for future LOA management. In the current version, LOA-related behaviour is mainly adaptable: the operator manually selects modes and commands. However, the same interface structure can be extended to support delegated and supervised configurations, in which the system presents candidate actions, registers planned

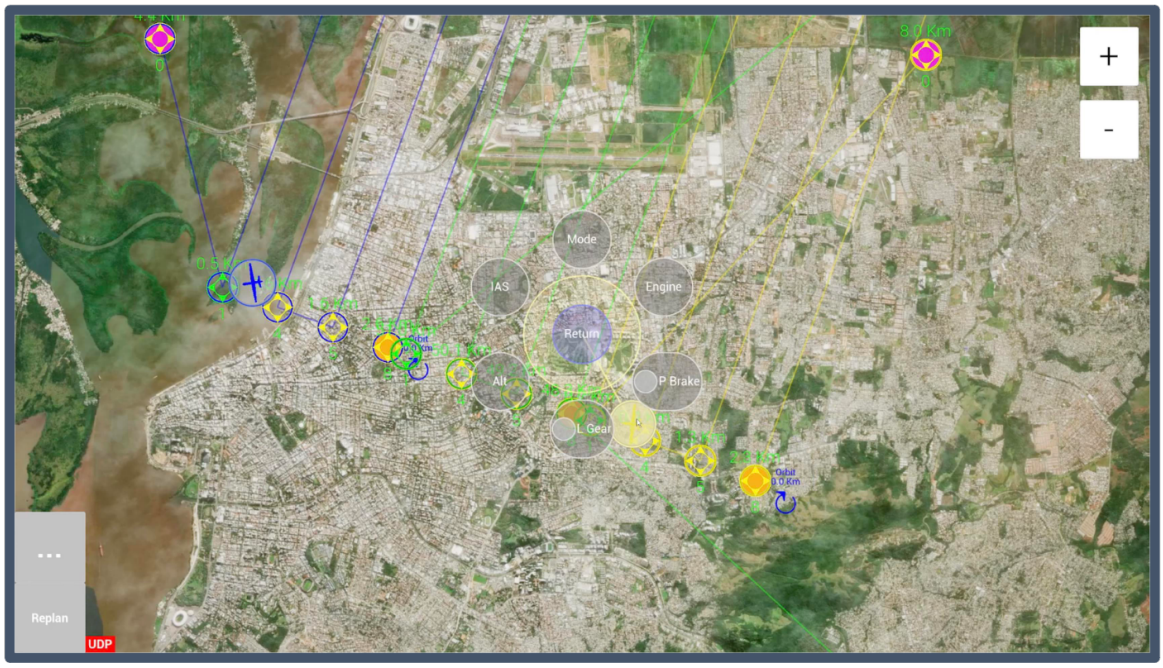


Figure 5 – V1 prototype: proposed map-based HMI for multi-UAV command and control.

actions, or recommends mode changes according to mission phase, failures, or operator-state estimates.

4.4 Integration Results

The V1 demonstration confirmed the feasibility of integrating the simulation, visualisation, and HMI components in a single runnable prototype. Three UAVs can be represented simultaneously in the visual environment, their camera views and telemetry can be displayed to the operator, and commands can be issued through the map-based interface. The result therefore validates the basic information flow required by the proposed co-simulation architecture: simulated vehicle states are transmitted to the visualisation layer, displayed through HMI elements, and operator commands are returned to the simulation environment.

The implemented prototype also confirms the feasibility of separating the system into modular components. The flight-dynamics and mission-management layers can evolve independently from the HMI layout and geospatial visualisation. Likewise, additional functions such as failure injection, mission re-planning, LOA recommendation, and workload estimation can be incorporated without redesigning the complete system architecture.

4.5 Evaluation Readiness

Although the current prototype does not yet include formal human-subject validation, it provides the infrastructure required for the next experimental stage. In particular, the system is prepared to record vehicle states, mission events, operator commands, interface states, and LOA-related actions using a common time base. This logging capability is essential for later comparison of LOA configurations, analysis of operator responses to failures, and correlation between mission performance and workload or situation-awareness measures.

The V1 prototype therefore establishes the operational basis for future human-in-the-loop experiments. In the next development stage, the implemented HMI will be extended with systematic failure injection, re-planning alternatives, explicit LOA-selection workflows, and workload-related indicators. These additions will allow the comparison of human-operated, human-delegated, and human-supervised configurations under nominal and degraded mission conditions.

5. Conclusion

This paper presented a methodology and first prototype implementation for an adaptive/adaptable HMI applied to single-operator, multi-UAV supervision in a flood SAR context. The proposed methodology provides a traceable development process that links operational scenario definition, task analysis, function decomposition, LOA, adaptation-policy design, interface controls, transparency mechanisms, and evaluation variables. In this way, the HMI is treated not only as a visual interface, but as an operational component that mediates the interaction between the operator, the UAV fleet, the mission-management logic, and the autonomy-management layer.

The co-simulation architecture was designed to support this methodology in a repeatable experimental environment. The framework integrates flight-dynamics simulation, mission-management logic, real-time visualisation, operator interaction, and synchronised logging. This structure allows UAV states, mission events, LOA changes, operator commands, and interface states to be represented consistently, creating the basis for future comparisons between different autonomy configurations and HMI strategies.

The V1 prototype demonstrated the feasibility of implementing the proposed approach in a runnable multi-UAV scenario. The prototype includes a georeferenced flood-response environment, simultaneous monitoring of three fixed-wing UAVs, sensor-camera views with telemetry overlays, and a map-based command-and-control interface with waypoint, orbit, flight-mode, speed, altitude, and basic vehicle commands. These results show that the proposed architecture can support both fleet-level monitoring and vehicle-level supervision while preserving explicit operator authority through adaptable controls.

The main limitation of the current version is that it does not yet report formal human-subject validation or closed-loop physiological adaptation. Therefore, the present results should be interpreted as implementation and integration results rather than as evidence of workload reduction or performance improvement. Nevertheless, the V1 prototype establishes the required infrastructure for the next experimental stage, including logging of vehicle states, mission events, operator interactions, and interface states.

Future work will extend the prototype by incorporating systematic failure injection, mission re-planning alternatives, explicit LOA-selection workflows, and workload-related indicators. These additions will enable controlled human-in-the-loop experiments comparing human-operated, human-delegated, and human-supervised configurations under nominal and degraded mission conditions. The expected outcome is a set of empirical results and design guidelines for selecting and adapting LOA in single-operator, multi-UAV SAR missions.

6. Acknowledgements

Thanks to the Aeronautics Institute of Technology (ITA), in particular, the Competence Center in Manufacturing (CCM), for granting the necessary facilities and structure for the development of the work and the financial support of the following Brazilian Funding Agencies: FAPESP with grant 2021/11258-5; FINEP with grants 01.24.0600.00, 01.20.0195.00 and 01.22.0478.00, CNPq and CISB.

7. Contact Author Email Address

The contact author's email address, Andrew Sarmento: andrewgps@ccm-ita.org.br

8. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Ben Grindley, Katie Phillips, Katie J. Parnell, Tom Cherrett, James Scanlan, and Katherine L. Plant. Over a decade of uav incidents: A human factors analysis of causal factors. *Applied Ergonomics*, 121:104355, 2024.

[2] Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. A model for types and levels of human interaction with automation. *IEEE Trans. on Systems, Man, and Cybernetics–Part A*, 30(3):286–297, 2000.

[3] M.R. Endsley. *Designing for Situation Awareness: An Approach to User-Centered Design*. CRC Press, second edition (2nd ed.) edition, 2011.

[4] Thomas A. O'Neill, Nathan J. McNeese, Amy Barron, and Beau Schelble. Human–autonomy teaming: A review and analysis of the empirical literature. *Human Factors*, 64(5):904–938, 2022.

[5] Jonas Lundberg, Mattias Arvola, and Karljohan Lundin Palmerius. Human autonomy in future drone traffic: Joint human–ai control in temporal cognitive work. *Frontiers in Artificial Intelligence*, 4:704082, 2021.

[6] The MathWorks Inc. MATLAB version: 24.2 (R2024b). `https://www.mathworks.com`, 2024.

[7] JSBSim Development Team. Jsbsim: An open source flight dynamics and control software library in c++. `https://github.com/JSBSim-Team/jsbsim`, 2025. GitHub repository; accessed 2025-11-24.

[8] Epic Games. Unreal engine 5 offers significant new potential for the simulation industry. `https://www.unrealengine.com/en-US/blog/unreal-engine-5-offers-significant-new-potential-for-the-simulation-industry`, 2022. Accessed: 24 Nov. 2025.

[9] Cesium GS, Inc. Cesium ion: 3d geospatial tiling and streaming platform. `https://cesium.com/platform/cesium-ion/`, 2025. Accessed: 24 Nov. 2025.

[10] G. Luzzani, I. Buraioli, D. Demarchi, and G. Guglieri. A review of physiological measures for mental workload assessment in aviation: A state-of-the-art review of mental workload physiological assessment methods in human–machine interaction analysis. *The Aeronautical Journal*, 128:928–949, 2024.

[11] Gaganpreet Singh, Caroline P. C. Chanel, and Raphaëlle N. Roy. Mental workload estimation based on physiological features for pilot-UAV teaming applications. *Frontiers in Human Neuroscience*, 15:692878, 2021.

[12] Xiaoli Fan, Chaoyi Zhao, Xin Zhang, Hong Luo, and Wei Zhang. Assessment of mental workload based on multi-physiological signals. *Technology and Health Care*, 28(S1):S67–S80, 2020.

[13] Stephen H. Fairclough. Fundamentals of physiological computing. *Interacting with Computers*, 21(1–2):133–145, 2009.

[14] Nichakorn Pongsakornsathien, Yixiang Lim, Alessandro Gardi, Samuel Hilton, Lars Planke, Roberto Sabatini, Trevor Kistan, and Neta Ezer. Sensor networks for aerospace human-machine systems. *Sensors*, 19(16):3465, 2019.

[15] Zhe Zhao, Yifeng Niu, and Lincheng Shen. Adaptive level of autonomy for human-UAVs collaborative surveillance using situated fuzzy cognitive maps. *Chinese Journal of Aeronautics*, 33(11):2835–2850, 2020.

[16] Yixiang Lim, Nichakorn Pongsakornsathien, Alessandro Gardi, Roberto Sabatini, Trevor Kistan, Neta Ezer, and Daniel J. Bursch. Adaptive human-robot interactions for multiple unmanned aerial vehicles. *Robotics*, 10(1):12, 2021.

[17] Pietro Aricò, Gianluca Borghini, Gianluca Di Flumeri, Alessandro Colosimo, Simone Bonelli, Alessia Golfetti, Silvia Pozzi, Jean-Philippe Imbert, Guillaume Granger, Rabah Benhacene, and Fabio Babiloni. Adaptive automation triggered by EEG-based mental workload index: A passive brain–computer interface application in realistic air traffic control environment. *Frontiers in Human Neuroscience*, 10:539, 2016.

[18] Gianluca Di Flumeri, Francesca De Crescenzio, Bruno Berberian, Oliver Ohneiser, Jan Kramer, Pietro Aricò, Gianluca Borghini, Fabio Babiloni, Sara Bagassi, and Sergio Piastra. Brain–computer interface-based adaptive automation to prevent out-of-the-loop phenomenon in air traffic controllers dealing with highly automated systems. *Frontiers in Human Neuroscience*, 13:296, 2019.

[19] Summer Rebensky, Kendall Carmody, Cherrise Ficke, and Sean D. Hutchins. Teammates instead of tools: The impacts of level of autonomy on mission performance and human–agent teaming dynamics in multi-agent distributed teams. *Frontiers in Robotics and AI*, 9:782134, 2022.

[20] Gang Huang, Min Hu, Xueying Yang, Xun Wang, Yijun Wang, and Feiyao Huang. A review of constrained multi-objective evolutionary algorithm-based unmanned aerial vehicle mission planning: Key techniques and challenges. *Drones*, 8(7):316, 2024.

[21] Agência Senado. MP prevê R$ 5 bi de crédito extra para mudanças climáti-

cas no RS. https://www12.senado.leg.br/noticias/materias/2024/10/23/mp-preve-r-5-bi-de-credito-extra-para-mudancas-climaticas-no-rs, October 2024. Senado Notícias. Acessado em: 15 jun. 2026.