A Model for Mediating Multi-Modal Human Intent into Safe Maneuvers for UAVs

Sofia Nelson
University of Notre Dame
Notre Dame, IN, USA
snelso24@nd.edu

Dalal Alrajeh
Imperial College London
London, UK
dalal.alrajeh@ic.ac.uk

Pedro Antonio Alarcon Granadeno
University of Notre Dame
Notre Dame, IN, USA
palarcon@nd.edu

Jane Cleland-Huang
University of Notre Dame
Notre Dame, IN, USA
janehuang@nd.edu

***Abstract*—Direct human interaction with autonomous UAV systems can be enabled through modalities such as speech, gestures, and graphical interfaces. However, interpreting such inputs as directly executable commands introduces safety risks in dynamic environments. Operator requests may conflict with terrain constraints, inter-UAV separation requirements, or flight-envelope limitations. In this paper, we present a requirements-governed maneuver-response model that mediates multi-modal human intent into safe UAV maneuvers by treating operator inputs as bounded maneuver requests rather than direct commands. Requested maneuvers are mapped to constrained motion primitives and processed through a structured request–evaluate–execute pipeline. Each request is interpreted with associated confidence, validated against terrain, separation, workspace, and flight-envelope constraints, and either constrained, rejected, or executed under continuous runtime monitoring. We further formalize the approach as a requirements-based specification model in which maneuver primitives are associated with explicit preconditions, invariants, guard conditions, and postconditions governing admissibility, execution safety, and emergency handling. These requirements support runtime verification and future reactive synthesis approaches. We present an initial lab-based validation demonstrating that voice- and GUI-based inputs can be reliably interpreted and safely executed as constrained maneuver requests.**

***Index Terms*—Human–UAV Interaction, Runtime Assurance, Safety-Critical Systems, Multi-Modal Interaction, Requirements**

I. Motivation

Within the Human-Machine Teaming (HMT) paradigm, small Uncrewed Aerial Vehicles (UAVs) operate as collaborative partners that combine autonomous decision-making with human oversight and intervention [1]. Although these systems are often deployed under human-on-the-loop (HoTL) supervision, operators may periodically engage more directly in the loop by issuing maneuver- or mission-level commands through speech, gestures, or hardware and software interfaces. Such interventions are especially important in dynamic operational contexts where human judgment, situational awareness, or mission intent may temporarily supersede autonomous behavior. For instance, an operator might direct a drone to reposition itself for a medical package delivery or to conduct a closer inspection of a structure when environmental conditions, mission priorities, or nuanced situational cues cannot be reliably interpreted by the autonomous system alone.

To support these forms of interaction, a significant body of work has explored mechanisms through which humans provide directives to UAVs, especially rotorcraft platforms capable of hovering and maneuvering with high precision in close proximity to people and objects. Several researchers have developed gesture-based systems, where real-time hand poses are mapped to flight commands for single UAVs [2]–[4] or even swarms [5], [6]. Others have used voice commands to enable hands-free communication [7], [8], and more recently multimodal fusion of gesture and voice inputs has been shown to outperform either modality alone, with each communication channel compensating for the other's environmental limitations [9], [10]. However, this prior work has focused primarily on using Computer Vision (CV) or Natural Language Processing (NLP) to recognize human intent and to map it to UAV actions, with the underlying assumption that human-issued commands can be executed safely and directly. This introduces very real safety concerns, as humans may issue incorrect or unclear commands due to limited visibility, loss of spatial awareness, distractions, or reckless behavior, causing the UAV to fly into the terrain, crash into an operator on the ground, or collide with another UAV in the air. The problem is further exacerbated because research experiments are often conducted in controlled or synthetic environments [9], [11], [12], where simplified conditions and limited environmental variability may obscure safety risks and overestimate how well these approaches generalize to real-world operational settings.

Despite the significant safety risks associated with direct command execution, existing safety-oriented work has primarily focused on issues such as human comfort, physical separation distances, and safe operating envelopes during human-UAV interaction rather than on the problem of constraining and mediating operator-issued commands prior to execution [7]. Addressing this problem requires more than accurate recognition of operator intent; it requires runtime mechanisms capable of determining whether a requested maneuver remains admissible under current environmental conditions, vehicle capabilities, mission objectives, airspace constraints, and team context. This introduces a runtime requirements engineering challenge in which operator inputs must be treated not as unconditional commands, but as high-level intent requests subject to continuous constraint evaluation and safety monitoring.

In this paper, we address these safety concerns by introducing a constrained multimodal operating model in which operator inputs are interpreted as requests for maneuver primitives rather than as direct actuation commands. Each requested maneuver is evaluated against terrain, separation, and flight-envelope constraints prior to execution and continuously monitored during execution. Beyond the operational model itself, we formulate the command mediation process as a requirements-based specification model in which maneuver primitives are associated with explicit preconditions, invariants, guard conditions, and postconditions governing admissibility, execution safety, and emergency handling. These requirements provide a foundation for runtime verification and future reactive synthesis approaches while preserving intuitive human interaction with the autonomous system.

The remainder of the paper is laid out as follows. Section II describes related work. Section III presents the maneuver-response model, including bounded maneuver primitives, admissibility and safety constraints, the request–evaluate–execute pipeline, and runtime monitoring mechanisms, while Section IV then describes the requirements-governed operationalization of the model through safety properties, runtime guards, state-machine behavior, and illustrative maneuver contracts. Section V reports on preliminary experiments and their results. The paper concludes in Sections VI and VII with a discussion of threats to validity, conclusions, and future work.

II. Related Work

Related work primarily falls under the topics of multimodal inputs for UAV controls, including speech, gestures, and hardware or software interfaces.

A. Multi-Modal Implementation

Multimodal approaches combine modalities such as speech and hand gestures to create more flexible UAV control systems. Although gestures are not strictly hands-free, they reduce reliance on handheld controllers or screen-based input, enabling more natural UAV interaction in field settings [7], [8], and both approaches have been shown to outperform traditional joystick interfaces [2], [13]. Prior work has demonstrated that combining modalities can improve performance as each performs better under different environmental conditions [9], [10]. For example, speech may remain effective in low-light conditions, whereas gestures can be more reliable in noisy environments [9]. Multi-modal systems also enable cross-modal confirmation, where one modality can help validate another and reduce misinterpretation [9], [10]. However, prior evaluation has primarily been in controlled indoor or synthetic environments without addressing the variability of real-world operations [9], [12]. As a result, deployment in complex operational settings has been limited, and explicit runtime safety constraint mechanisms have not been rigorously addressed [9], [12], [14].

B. Speech Recognition

Voice recognition offers an intuitive alternative to traditional remote-control and tactile UAV interfaces and can be effectively integrated into UAV control pipelines through automatic speech recognition (ASR) frameworks [15]. However, environmental noise and signal distortion remain significant challenges. These issues become even more pronounced in collaborative operational environments, where robust recognition may require techniques such as hidden Markov models (HMMs), deep neural network (DNN) acoustic models, grammar-based syntax analysis, and semantic post-processing to reduce ambiguity in safety-critical settings [16]. More recently, researchers have explored the use of large language models (LLMs) in UAV speech-control pipelines to support more flexible natural-language interaction [17]. These systems can translate free-form and spoken instructions into executable UAV commands using hybrid STT and LLM-based architectures [17]. Although promising, current approaches still rely heavily on cloud-based inference and remain largely evaluated in controlled or simulated environments, leaving robust real-world deployment as an open challenge [17].

C. CV Gesture Recognition

CV techniques provide an alternative approach for real-time recognition of human motions and gestures. While early approaches relied on explicit feature extraction, recent work has used deep-learning techniques to provide faster and more robust recognition [4], [18]–[20]. Depth-sensing approaches capture three-dimensional positional information [21]–[23], while modern convolutional neural network (CNN) methods learn spatial-temporal features directly from images and video streams [24]–[26]. However, these systems remain sensitive to environmental factors such as lighting and occlusion [27], and models trained on controlled datasets often fail to generalize to real-world environments [11], [28]. Gesture-recognition systems have been applied to UAV control by mapping hand poses to commands such as takeoff, landing, and directional movement [29]. These systems often rely on small gesture vocabularies and controlled datasets [29]. Gesture control has also been extended to UAV swarms, where a single operator may coordinate multiple drones simultaneously [5], [6]. However, scalability and safety remain major challenges, since a single misinterpreted command may affect multiple UAVs at once [5], [6]. While frameworks such as ROS2 support scalable implementations [30], current approaches still provide limited support for uncertainty propagation and runtime safety constraint evaluation [31]. Consequently, system reliability depends not only on recognition accuracy, but also on how failures and ambiguities are handled at the control level [32], [33].

D. The Need for Runtime Safety Mediation

Taken together, these multimodal approaches demonstrate significant promise for enabling intuitive and flexible UAV interactions. However, speech, gesture, and GUI inputs remain inherently uncertain and susceptible to misinterpretation due

to environmental conditions, ambiguity, sensor limitations, and human error. Consequently, reliable human-UAV interaction depends not only on accurate intent recognition, but also on runtime mechanisms capable of evaluating whether requested maneuvers remain safe and admissible within the current operational context [1], [34]–[36].

III. Multi-Modal Maneuver Request Model

In this section we present our Multi-Model Maneuver Request (M3R) model. At a high-level, the UAV is always in one of the following modes.

- **NormalAutonomy**: The drone performs its mission under nominal autonomous control. This high-level mode encompasses a variety of lower level states such as takeoff, land, fly-to-waypoints, and circle. In this mode the drone is **disengaged** from direct interaction with the human operator.
- **ManeuverResponseArmed**: The drone is eligible to receive maneuver commands, but no command is currently being acted upon.
- **ManeuverResponseActive**: A maneuver request has been accepted and translated into a bounded motion primitive.
- **Blocked**: A requested maneuver has been rejected because it would violate one or more safety constraints.
- **EmergencyHold**: The UAV has interrupted execution of an MRM maneuver and transitioned to a safe hover or hold state.
- **ReturnToAnchor**: The drone returns to a safe anchor point after a maneuver or interruption.

A. Runtime Pairing of Operator and UAV

Pairing occurs between a single operator and a single UAV when the operator sends a *pairing request* over existing communication channels (e.g., telemetry or mesh-radio). The request contains the operator's coordinates (latitude, longitude, and altitude above mean sea level (AMSL)). After receiving the request, the UAV rotates to face the operator, transitions into the *ManeuverResponseArmed* state, and returns a confirmation message that may include visual or auditory cues in addition to standard communication mechanisms. As the UAV is required to maintain an operator-facing orientation, all subsequent directional commands are interpreted relative to the operator's perspective rather than the UAV body frame. For example, if the operator instructs the UAV to "move right," the UAV moves to its left. This design choice reduces ambiguity during real-world operations, particularly at longer distances where operators may struggle to accurately perceive the UAV's heading and orientation.

B. Mapping Maneuver Commands to Primitive Actions

The M3R model allows the UAV to act upon a finite set of requests, each of which is mapped to a bounded motion primitive. The current set of mappings are depicted in Table I.

Each primitive includes the following attributes:

- intended direction,
- maximum displacement,
- maximum speed,
- maximum duration,
- completion condition,
- safety preconditions, and
- abort conditions.

C. Request–Evaluate–Execute Pipeline

Each maneuver request passes through the following stages:

1) **Interpret**: Process the input request (e.g., gesture, speech, or GUI interaction), compute confidence in the interpretation of the request, and if confidence is sufficient, infer the intended maneuver. We establish a *confidence-threshold* (cf) for all examples and experiments reported in this paper.
2) **Validate**: Check whether the requested maneuver is allowed in the current context. For example, if the operator requests the UAV to *move-right*, but there is a steep incline in that direction, the UAV will tag the request as *unsafe*.
3) **Constrain**: In the case of an unsafe maneuver, the UAV either prohibits the maneuver or reduces it to a safe bounded version.
4) **Execute**: Perform the approved motion primitive.
5) **Monitor**: Continuously evaluate safety constraints during execution.
6) **Complete or Abort**: Terminate the maneuver when complete or interrupt it if a violation is detected.

TABLE I: Baseline maneuver requests are mapped to bounded intentions. The multimodal requests can be provided by gestures, verbal commands, or through a GUI.

Maneuver Request	Primitive Action	Nominal Intent	Example Bounds
Go left	MoveLeft	Lateral translation left	$\leq 1\,\mathrm{m} \wedge (velocity < x\,\mathrm{m/s})$
Go right	MoveRight	Lateral translation right	$\leq 1\,\mathrm{m} \wedge (velocity < x\,\mathrm{m/s})$
Go forward	MoveForward	Forward translation	$\leq 1\,\mathrm{m} \wedge (velocity < x\,\mathrm{m/s})$
Go back	MoveBackward	Rearward translation	$\leq 1\,\mathrm{m} \wedge (velocity < x\,\mathrm{m/s})$
Go up	Ascend	Vertical translation upward	$\leq 0.5\,\mathrm{m} \wedge (velocity < x\,\mathrm{m/s})$
Go down	Descend	Vertical translation downward	$\leq 0.5\,\mathrm{m} \wedge (velocity < 0.5x\,\mathrm{m/s})$
Turn left/right	ControlledYawScan	Slow full rotation, return to operator-facing orientation	$= 360^\circ \wedge (\dot{\psi} < \omega_{\mathrm{scan}}\,\mathrm{deg/s}) \wedge \psi_{\mathrm{final}} = \psi_A$

D. Anchor-Based Local Workspace

When the *ManeuverResponseArmed* mode is initially activated, the UAV records its current position and attitude as its *anchor state*:

$$A = (x_A, y_A, z_A, \psi_A)$$

where x_A, y_A, and z_A transform latitude, longitude, and altitude into a local coordinate systems, and ψ_A represents the direction the UAV is facing (heading). Notably this must be towards the operator with which the UAV is paired. Each subsequent maneuver is constrained to remain within a bounded local workspace around this anchor, defined as

$$(x - x_A)^2 + (y - y_A)^2 + (z - z_A)^2 \leq R_{\max}^2$$

where, for example $R_{\max} = 5\,\mathrm{m}$.

This workspace may be further clipped by adjacent terrain, nearby drones, or geofencing constraints. Each time a new command is received by the UAV, it dynamically updates its anchor and subsequently the boundaries of its allowed workspace.

E. Safety Constraint Categories

In addition to providing assurances that the UAV remains within its bounded local workspace, the safety model evaluates each requested maneuver against four complementary categories of constraints derived from common UAV failure modes and operational risks (e.g., [37]).

• **Terrain Constraints:** Terrain constraints ensure that the UAV maintains safe clearance relative to the ground and local topography. Example conditions include:

$$\mathrm{AGL}(s') \geq \mathrm{MinAGL}$$
$$\mathrm{TerrainClearance}(\pi) \geq C_{\min}$$

where s' is the predicted future state and π is the predicted path for the requested maneuver. The first constraint ensures that the UAV does not descend below a minimum allowable altitude above ground level, while the second ensures that the predicted maneuver path maintains sufficient clearance from terrain and obstacles.

• **Inter-UAV Separation Constraints:** Inter-drone constraints ensure safe spacing and prevent conflicts with nearby drones. For each neighboring drone $d \in D$:

$$\mathrm{Sep3D}(\pi_{self}, \pi_d) \geq S_{\min}$$

where π_{self} is the predicted path of the requesting drone and π_d is the predicted path or reserved path volume of drone d. This constraint ensures that the minimum three-dimensional separation between the two predicted trajectories remains above a required safety threshold throughout maneuver execution.

• **Flight Envelope Constraints:** Flight envelope constraints bound the magnitude, velocity, and rotational rate of each maneuver:

$$|\Delta x| \leq X_{step}$$
$$|\Delta y| \leq Y_{step}$$
$$|\Delta z| \leq Z_{step}$$
$$v \leq V_{\max}^{nudger}$$
$$\dot{\psi} \leq \Omega_{\max}^{nudger}$$

where Δx, Δy, and Δz represent the requested positional displacement along each axis, v denotes translational velocity, and $\dot{\psi}$ denotes yaw rate. These constraints ensure that MR interactions remain bounded supervisory maneuvers rather than unconstrained teleoperation.

• **Context and Confidence Constraints:** Context and confidence constraints determine whether the UAV transitions from *ManeuverResponseArmed* to *ManeuverResponseActive* mode under current operating conditions:

$$\mathrm{InputConfidence}(m) \geq C_{\min}^{(m)}$$
$$\mathrm{LocalizationConfidence} \geq L_{\min}$$
$$\mathrm{TerrainModelConfidence} \geq T_{\min}$$
$$\mathrm{NeighborStateAge} \leq \tau_{\max}$$

where m denotes the input modality (e.g., gesture, speech, or GUI), $\mathrm{InputConfidence}(m)$ represents the confidence associated with the interpreted request from that modality, and $\mathrm{NeighborStateAge}$ denotes the elapsed time since the most recent state update for nearby agents. These constraints ensure that maneuvers are only permitted when the system possesses sufficiently reliable input interpretation, localization accuracy, terrain awareness, and neighboring drone state information. A maneuver is blocked if any condition is violated.

F. Safety Decision Function

When a UAV receives a maneuver request, it analyzes the request to determine whether the maneuver can be executed safely within the current operational context. The UAV performs this analysis using the following safety-decision function. Let

- g denote the recognized MR,
- x denote the current UAV state,
- T denote the terrain model,
- D denote the set of neighboring UAVs, and
- C denote contextual confidence values.

The system computes:

$$\mathrm{intent} = \mathrm{Interpret}(g)$$
$$\pi_{cand} = \mathrm{Plan}(\mathrm{intent}, x)$$
$$\pi_{safe} = \mathrm{Filter}(\pi_{cand}, T, D, C)$$

A candidate maneuver is admissible only if

$$\mathrm{Safe}(\pi_{cand}, x, T, D, C) := \mathrm{TerrainSafe} \wedge$$
$$\mathrm{SeparationSafe} \wedge$$
$$\mathrm{EnvelopeSafe} \wedge$$
$$\mathrm{ConfidenceAdequate}.$$

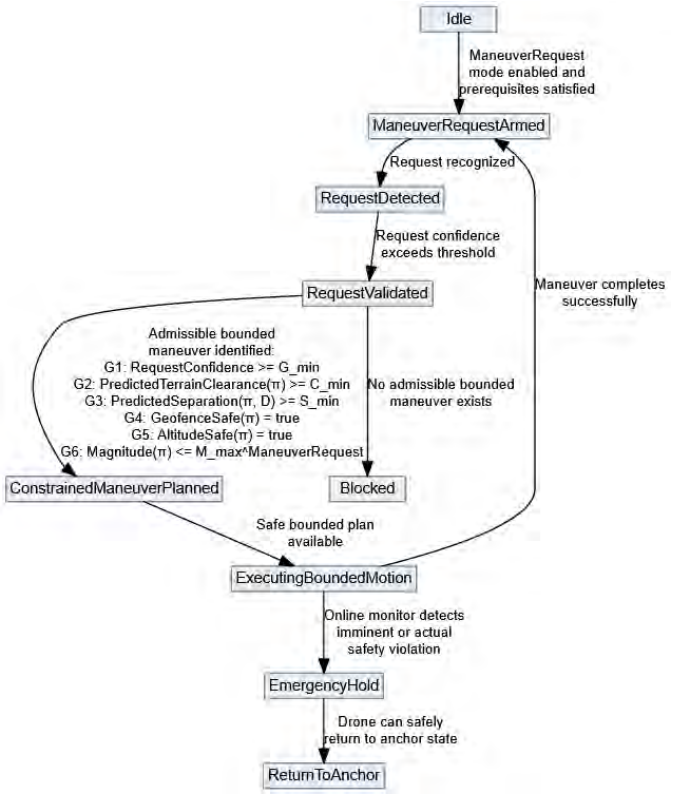

Fig. 1: State machine operationalizing the M3R model, including request interpretation, safety validation, constrained maneuver planning, execution, and runtime intervention.

If $\mathrm{Safe}(\pi_{cand}, x, T, D, C)$ is false, the maneuver is either reduced, delayed, rejected, or replaced with a safer alternative.

IV. Requirements-Guided Operationalization

The maneuver-response model presented in the previous section provides a conceptual framework for mediating human intent into safe UAV motion. In this section, we show how the model can be translated into implementation-oriented artifacts through a requirements-guided operationalization process. We first derive a formal requirements specification that captures the behavioral and safety properties of the maneuver-response model. We then present two complementary operationalizations of this specification: a state-machine representation describing system-level behavior and maneuver-specific contracts defining the requirements associated with individual maneuver primitives. Together, these artifacts provide a traceable path from high-level safety requirements to executable controller behavior while supporting verification and future automated controller synthesis.

A. Requirements Specification

The maneuver-response model is first expressed as a set of behavioral and safety requirements. These requirements are derived from the preconditions, invariants, guard conditions, and postconditions associated with each maneuver primitive, and constrain the UAV state, terrain model, operator input confidence, and neighboring UAVs. Collectively, they define the expected behavior of the maneuver-response system independently of any implementation.

At a high level, the specification enforces the following properties:

- *ManeuverResponseArmed* mode is activated only after successful pairing has occurred between the operator and the UAV.
- A maneuver request is validated only when the associated input confidence exceeds the required threshold and the candidate maneuver satisfies all applicable safety constraints.
- When an intended maneuver fails to satisfy spatial or environmental safety constraints, the requested maneuver is iteratively shortened along its original intended path until a safe bounded trajectory is identified. The maneuver is rejected if no admissible trajectory exists.

For example, the second requirements above (i.e., low-confidence requests must never activate maneuver execution) can be expressed formally in signal temporal logic [] as:

$$\square\Big((\mathrm{ManeuverRequestDetected} \\ \wedge\ \mathrm{GestureConfidence} < G_{\min}) \\ \rightarrow \bigcirc(\texttt{Blocked})\Big)$$

where $\square$ denotes the "always" operator and $\bigcirc$ denotes the "next-state" operator.

The resulting requirements provide a formal specification against which operational models and implementations can be evaluated. Requirements governing maneuver admissibility, safety constraints, workspace bounds, and emergency handling support systematic analysis, verification, testing, and traceability throughout the development process. They also establish a foundation for future requirements-driven controller synthesis, in which environmental assumptions and system guarantees may be used to automatically construct correct-by-design maneuver-response controllers. In the following subsection, we present a baseline state-machine realization that operationalizes this specification.

B. State Machine Representation

One operational realization of the requirements specification is a maneuver-response state machine. The state machine provides an executable behavioral model in which each transition corresponds to the satisfaction of one or more requirements defined in the previous subsection. Such a representation may be developed manually as part of the software engineering process or generated automatically through future controller synthesis techniques. A baseline realization is presented below and illustrated in Figure 1.

- `Idle` → `ManeuverRequestArmed` when MR mode is enabled and prerequisites are satisfied.
- `ManeuverRequestArmed` → `ManeuverRequestDetected` when a gesture is recognized.

- `ManeuverRequestDetected` → `ManeuverRequestValidated` when gesture confidence exceeds threshold.
- `ManeuverRequestValidated` → `ManeuverPlanned` when safety guards are satisfied.
- `ManeuverRequestValidated` → `Blocked` when one or more safety constraints fail.
- `ManeuverPlanned` → `ExecutingBoundedMotion` when a safe plan is available.
- `ExecutingBoundedMotion` → `EmergencyHold` when an online monitor detects imminent or actual safety violation.
- `ExecutingBoundedMotion` → `NudgerArmed` when the maneuver completes successfully.
- `EmergencyHold` → `ReturnToAnchor` when the drone can safely return to the anchor state.

Representative guard conditions for transition to `ManeuverPlanned` are:

$$G_1 : \text{GestureConfidence} \geq G_{\min}$$
$$G_2 : \text{PredictedTerrainClearance}(\pi) \geq C_{\min}$$
$$G_3 : \text{PredictedSeparation}(\pi, D) \geq S_{\min}$$
$$G_4 : \text{GeofenceSafe}(\pi) = \text{true}$$
$$G_5 : \text{AltitudeSafe}(\pi) = \text{true}$$
$$G_6 : \text{Magnitude}(\pi) \leq M_{\max}^{nudger}$$

These guard conditions correspond to runtime-monitorable signals and operationalize the behavioral and safety requirements defined in the previous subsection. Consequently, the state machine provides a concrete implementation model whose execution can be verified against the underlying requirements specification.

C. Contract-Based Operationalization

A complementary operationalization of the requirements specification is provided through maneuver-specific behavioral contracts. Rather than describing system-wide behavior as a state machine, contracts specify the obligations associated with individual maneuver primitives in terms of preconditions, invariants, and postconditions. The following example illustrates the contract associated with the `GoForward` maneuver.

GOFORWARD

Preconditions

- nudger mode is active,
- gesture confidence is above threshold,
- localization is healthy,
- terrain and neighbor state data are sufficiently fresh.

Invariants

- terrain clearance remains above minimum threshold,
- inter-drone separation remains above minimum threshold,
- motion remains inside the bounded nudger workspace,
- speed remains below the nudger-mode speed bound.

Postconditions

- the drone reaches an approved final state within the local workspace, or
- the maneuver is safely aborted to hover or hold.

These contracts provide a direct realization of the requirements specification at the maneuver level. Preconditions define when execution may begin, invariants capture the safety properties that must be preserved throughout execution, and postconditions characterize acceptable completion or safe termination. Together with the state-machine representation, they provide complementary operational models that can be verified against the common requirements specification while maintaining traceability from high-level safety requirements to implementation.

V. IMPLEMENTING THE MODEL: AN INITIAL PROTOTYPE

While our model lends itself to both formal verification and reactive control synthesis, for purposes of this workshop paper we have implemented it using a manual requirements-driven design and validation process.

This separation ensures that safety does not depend exclusively on perfect human input recognition.

A. Requirements

We translated the model into a baseline set of requirements which which were then used to design, implement, and validate the DR-Maneuvers for our input modalities. The requirements are specified as follows using Easy Requirements Specification (EARS) notation [38].

1) When an MR-Maneuver input is recognized, the Maneuver-Request Interpretation Layer shall interpret the input as a request for a bounded motion primitive.
2) When a maneuver request is interpreted, the MR-Planner shall generate a bounded candidate maneuver plan.
3) When a candidate maneuver plan is generated, the Safety Supervisor shall evaluate the plan against predicted terrain clearance before execution.
4) When execution of a maneuver plan would reduce terrain clearance below the configured minimum safe threshold, the Safety Supervisor shall reject or constrain the maneuver plan.
5) When a candidate maneuver plan is generated, the Safety Supervisor shall evaluate the plan against the predicted positions or reserved path volumes of nearby UAVs.
6) When a maneuver plan would violate minimum inter-UAV separation constraints, the Safety Supervisor shall reject or constrain the maneuver plan.
7) While ManeuverResponse mode is active, the MR-Planner shall constrain maneuver plans to remain within the bounded local workspace relative to the current anchor state.
8) While ManeuverResponse mode is active, the MR-Planner shall limit maneuver displacement, velocity, and duration to configured flight-envelope bounds.
9) While a maneuver is executing, the Runtime Monitor shall continuously monitor terrain clearance, inter-UAV separation, and workspace-bound violations.

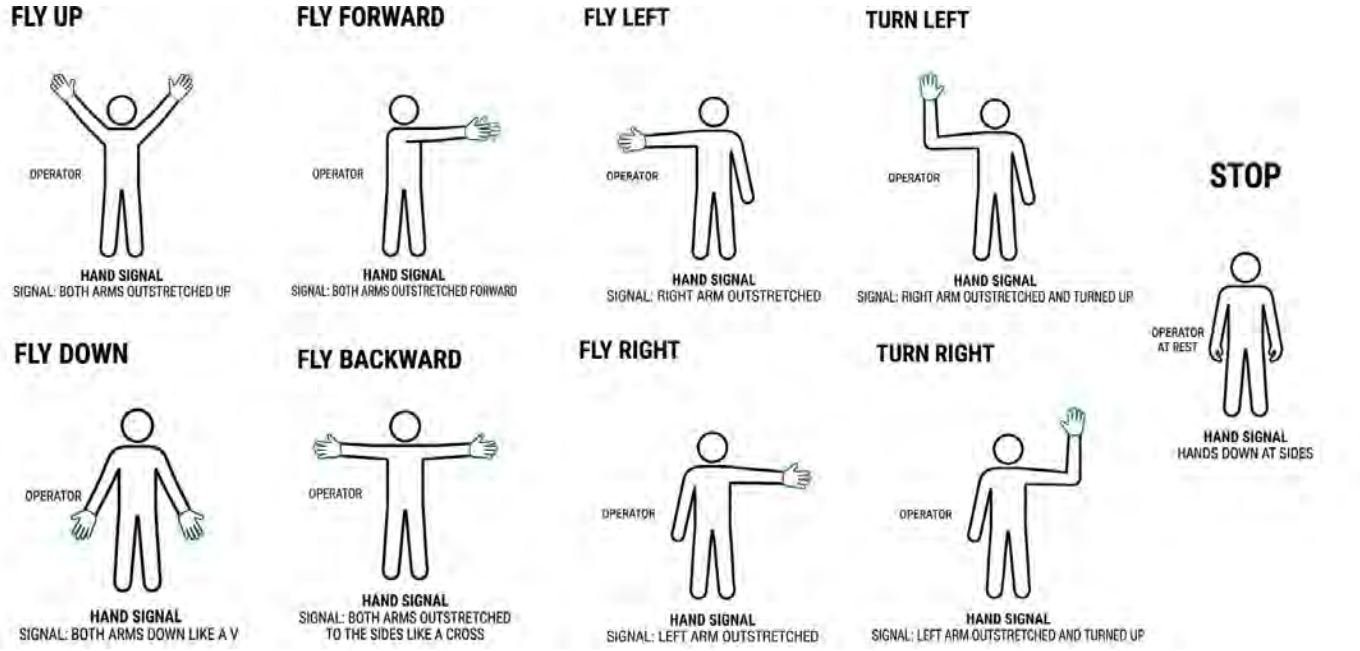

Fig. 2: Motion primitives for gesture-based commands. Each command supports "Fly" or "Go" prefix, but are shown here as 'Fly' commands.

10) When a hard safety constraint is violated or predicted to be violated during maneuver execution, the Runtime Monitor shall command the Motion Controller to abort the maneuver and transition the UAV into the EmergencyHold state.
11) When localization confidence, input confidence, terrain-model confidence, or neighboring-UAV state freshness falls below required thresholds, the Safety Supervisor shall reject or constrain the requested maneuver.

B. Architecture

The prototype architecture separates out input detection and interpretation from safety enforcement. A baseline decomposition includes:

- *Maneuver-Request Interpretation Layer*: recognizes gestures, speech, and GUI inputs, and infers requested maneuvers,
- *MR-Planner*: converts intentions into local bounded candidate plans,
- *Safety Supervisor*: evaluates plans against terrain, drone separation, geofencing, and confidence constraints,
- *Motion Controller*: executes only approved maneuvers, and
- *Runtime Monitor*: continuously checks safety invariants during execution.

C. Modality Interfaces

We have designed initial interfaces for all three modalities and implemented prototypes that have undergone preliminary tests in a lab environment with a simple simulated drone.

1) Gesture-Based Maneuvers: Our current design supports nine unique commands as described in Table I. We recognize all nine gestures in the current model but limit execution to Fly/Go Up, Down, Forward, Left, Right, Backward, Forward, and Stop. While Turn Left and Turn Right are conceptually simple to enact, they require additional work (outside the scope of this workshop paper) to assure that (a) when operating purely in gesture mode the drone always returns to face the operator, or (b) when operating in multimodal input mode (e.g., Gesture + Voice) commands such as 'Go Left' are enacted relative to the operator regardless of direction the drone is facing during the turn. For purposes of the initial prototype we used the Visual Large Language Model (VLMM) capabilities of GPT 5.o to recognize each of the human gestures in a video stream. This will be replaced in later work with a Yolo-trained model to recognize gestures at distance and altitude in potentially inclement conditions using our NOMAD data set [39].

2) Voice-Based Maneuvers: Voice activated maneuvers are recognized using VOSK, an open-source, offline speech recognition toolkit built on the Kaldi speech recognition framework. VOSK uses acoustic models trained with Kaldi's toolchain, which is based on deep neural networks for acoustic modeling combined with n-gram language models for decoding. The chosen model is `vosk-model-en-us-0.22`, a mid-sized (1.8GB) model trained on American English acoustic data.

The audio is collected from a microphone, and is captured continuously at 16 kHz, mono, 16-bit PCM in chunks of 4000 frames (∼250ms each), which are processed individually using VOSK's `KaldiRecognizer`. The recognizer is initialized upfront with a JSON grammar list comprised of every "legal"

(a) Go Back (b) Go Forward (c) Go Right

Fig. 3: A sequence of motion commands issued by human operators in an outdoor setting, and tested for recognition by the VLLM. All commands are from the user's perspective, therefore Go Right is executed by the drone moving to its left as it faces the human.

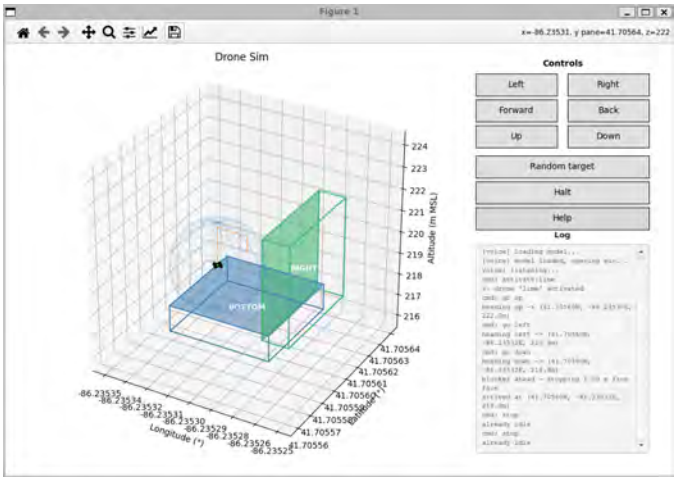

Fig. 4: Our prototype tool implements Speech, Gestures, and GUI-based commands interchangeably. The model updates its anchor position upon completion or interruption of each command, and halts at a predefined distance from terrain barriers (shown here in blue and green).

phrase, constraining the acoustic model to only consider known commands as candidates and labeling all non-relevant tokens as `[unk]`. The recognizer collects *partial results* as audio arrives, and then once silence is detected, it outputs *final results* matching a term in the grammar. NLP cleaning is applied to filter out noise, and then three sequential pattern checks are run in priority order:

1) Check for activation phrases ("ACTIVATE color" or "DISCONNECT color" substrings)
2) Check for "STOP/CONTINUE" commands
3) Check for directional phrases, which map every natural variant ("FLY BACKWARDS, FLY BACK, GO BACKWARD", etc.) to a single canonical direction (e.g., "`back`"). The first match "wins," and is returned. If nothing matches, `None` is returned, and the utterance is discarded.

Given this implementation, *Voice Activation Maneuvers* are only generated in cases when confidence is high that they match a targeted key-term.

D. Graphical User Interface Inputs

Our third modality involves a simple User Interface with buttons representing each of the actions shown in Figure 2. In future implementations we will add addition hardware interfaces such as a joystick.

E. Multi-Model Inputs

Finally, we built a prototype tool which accepts inputs from any of our three modalities interchangeably. The tool is shown in Figure 4. The tool was intended for initial validation purposes only. It was built using Python and MatPlotLib and provide a 3D environment defined by a moving window of latitude, longitude, and altitude ranges. The drone itself was represented by a 2D image situated with a sphere representing the workspace. Prior to receipt of a command, the drone was anchored at the center point of the sphere, and the subsequent command was executed relative to that point.

The operating constraints related to avoiding terrain are supported by an Environmental Digital Twin [40] and avoiding other UAVs by an air-leasing technology that guarantees that under nominal conditions all UAVs operate in unique airspace [41].

VI. Threats to Validity

There are several threats to validity to this work. First and foremost, we have presented the maneuver-response model as a foundation for both requirements-driven development and control synthesis; however, only the requirements-driven path has been translated into a functioning prototype. Our claims regarding control synthesis and formal verification are therefore grounded in the analyzability of the model rather than on a synthesized controller. Second, the framework reflects a bounded set of maneuver primitives and safety constraints; however, several aspects of the model require additional exploration. For example, fully addressing turn-based maneuvers, was out of scope of this work, as it introduces additional challenges related to maintaining operator-UAV communication during and after rotation, and adds additional implementation-level requirements for addressing new commands that might be issued during a turn.

A third threat is that for initial prototyping purposes, we opted to use a general-purpose Visual Large Language Model (VLLM) rather than a purpose-built CV model and tested gesture recognition under close-range, ideal conditions leaving gesture recognition performance under field conditions unvalidated. Moving forward, we will conduct a large-scale field-based data collection, and use the collected data to train a CV model with empirically defined capabilities and limitations for accurately recognizing gestures at varying distances and pitches [39].

Fourth, in this paper our focus was on constructing the basic MR-Maneuver model, rather than on rigorous real-world implementation and validation. The model therefore does not yet account for real-world uncertainties such as wind,

GPS drift, communication latency, rapidly changing airspace, or degraded visual and acoustic conditions [42]. In future work, we will extend the model to address these real-world phenomenon, and integrate both the synthesized controller and requirements-derived solutions with our existing terrain model [40]) and air-space coordination system [41].

Finally, a fundamental claim of our work is that, unlike prior gesture- and speech-based interfaces for UAV maneuvers, we explicitly address runtime safety concerns. However, our current validation remains limited, as it currently only partially addresses real-world phenomena such as geolocation uncertainties, and rigorous field-based evaluation involving both static obstacles (e.g., terrain and structures) and dynamic obstacles (e.g., other UAVs) is needed. At present, our validation has been limited to controlled prototype experiments where obstacles were dynamically introduced into a simulated environment, and the system demonstrated its ability to halt or constrain a human-initiated maneuver when a safety violation is detected. Despite these limitations, the current work establishes the core maneuver-response abstraction, safety mediation pipeline, and requirements framework needed to systematically reason about multimodal human intent prior to execution. We therefore view this work not as a completed operational solution, but as a foundational step toward runtime-assured and formally analyzable human–UAV interaction in complex operational environments.

VII. Conclusions

Taken together, the threats and limitations discussed in the previous section reinforce a central theme of this work that safe multimodal UAV interaction cannot depend solely on accurate intent recognition. Instead, speech, gestures, and GUI interactions must be mediated through explicit runtime safety constraints prior to execution. To address this problem, we introduced a maneuver-response framework in which operator inputs are interpreted as bounded maneuver requests subject to continuous evaluation against terrain, separation, localization, confidence, and flight-envelope constraints.

Beyond the operational model itself, this work establishes a requirements-governed foundation for runtime assurance and future controller synthesis for human–UAV interaction in complex operational environments. The framework formalizes maneuvers through requirements, safety contracts, runtime guards, and state-machine behavior, thereby supporting analyzability, verification, and future reactive synthesis approaches.

Future work will focus on addressing the primary threats and open challenges identified in this paper, particularly the problem of preserving runtime assurance and analyzability under increasingly realistic operational conditions. This includes integrating terrain-aware reasoning capabilities [40], scene-understanding pipelines, dynamic airspace-awareness mechanisms, ADS-B data, and DroneResponse's Air-Leaser framework for fine-grained airspace deconfliction [41] into the maneuver-response pipeline. While many of these operational capabilities have already been developed independently, combining them within a unified maneuver-response framework introduces significant challenges for controller synthesis, runtime monitoring, and safety assurance due to uncertainty in sensing, localization, communication, and environmental dynamics. Evaluation will therefore progress incrementally from controlled studies within our MOMUX environment [43] toward increasingly realistic outdoor field experiments involving physical UAVs operating under real-world terrain, airspace, sensing, and multi-UAV coordination constraints.

References

[1] J. Cleland-Huang, A. Agrawal, M. Vierhauser, M. Murphy, and M. Prieto, "Extending MAPE-K to support human-machine teaming," in International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2022, Pittsburgh, PA, USA, May 22-24, 2022, B. R. Schmerl, M. Maggio, and J. Cámara, Eds. ACM/IEEE, 2022, pp. 120–131. [Online]. Available: https://doi.org/10.1145/3524844.3528054

[2] M. Yoo, Y. Na, H. Song, G. Kim, J. Yun, S. Kim, C. Moon, and K. Jo, "Motion estimation and hand gesture recognition-based human–UAV interaction approach in real time," Sensors, vol. 22, no. 7, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/7/2513

[3] B. Hu and J. Wang, "Deep learning based hand gesture recognition and UAV flight controls," International Journal of Automation and Computing, vol. 17, 09 2019.

[4] S. Abdalla and S. Baidya, "UAV control with vision-based hand gesture recognition over edge-computing," 05 2025, pp. 481–488.

[5] M. Hu, J. Li, R. Jin, C. Shi, L. Xu, and R. Liu, "Hgic: A hand gesture based interactive control system for efficient and scalable multi- UAV operations," 2024. [Online]. Available: https://arxiv.org/abs/2403.05478

[6] R. S. Wijaya, S. Prayoga, R. A. Fatekha, and M. T. Mubarak, "A real-time hand gesture control of a quadcopter swarm implemented in the gazebo simulation environment," Journal of Applied Informatics and Computing, vol. 9, no. 3, p. 979–988, Jun. 2025. [Online]. Available: https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/9578

[7] Z. Zhu, J.-Y. Cheng, I. Jeelani, and M. Gheisari, "Using gesture and speech communication modalities for safe human-drone interaction in construction," Advanced Engineering Informatics, vol. 62, p. 102827, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034624004750

[8] S. Pourmehr, V. M. Monajjemi, R. Vaughan, and G. Mori, ""you two! take off!": Creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands," in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 137–142.

[9] A. Abioye, "Multimodal speech and visual gesture control interface technique for small unmanned multirotor aircraft," Ph.D. dissertation, University of Southampton, July 2023. [Online]. Available: https://eprints.soton.ac.uk/479472/

[10] X. Xiaojia, Q. Tan, H. Zhou, D. Tang, and J. Lai, "Multimodal fusion of voice and gesture data for UAV control," Drones, vol. 6, p. 201, 08 2022.

[11] I. Guyon, V. Athitsos, P. Jangyodsuk, and H. J. Escalante, "The chalearn gesture dataset (cgd 2011)," Mach. Vision Appl., vol. 25, no. 8, p. 1929–1951, Nov. 2014. [Online]. Available: https://doi.org/10.1007/s00138-014-0596-3

[12] J. Hu, "Integrated UAV swarm and human-machine interaction platform for real-time mobile training and evaluation," International Journal of Interactive Mobile Technologies (iJIM), vol. 19, pp. 83–97, 12 2025.

[13] C. Villame and S. Guirnaldo, "Design and implementation of voice-command controller for fixed-wing unmanned aerial vehicles using automatic speech recognition and natural language processing techniques," Sustainable Engineering and Innovation, vol. 6, pp. 199–212, 10 2024.

[14] A. Zhou, L. Han, and Y. Meng, Multimodal Control of UAV Based on Gesture, Eye Movement and Voice Interaction, 01 2023, pp. 3765–3774.

[15] R. Contreras, A. Ayala, and F. Cruz, "Unmanned aerial vehicle control through domain-based automatic speech recognition," Computers, vol. 9, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2073-431X/9/3/75

[16] J.-S. Park and N. Geng, "In-vehicle speech recognition for voice-driven UAV control in a collaborative environment of MAV and UAV," Aerospace, vol. 10, no. 10, 2023. [Online]. Available: https://www.mdpi.com/2226-4310/10/10/841

[17] K. Choutri, S. Fadloun, A. Khettabi, M. Lagha, S. Meshoul, and R. Fareh, "Leveraging large language models for real-time UAV control," Electronics, vol. 14, no. 21, 2025. [Online]. Available: https://www.mdpi.com/2079-9292/14/21/4312

[18] T. Starner and M. Group, "Visual recognition of american sign language using hidden markov models," 05 1995.

[19] C. Cui, M. S. Sunar, and G. Su, "Deep vision-based real-time hand gesture recognition: a review," PeerJ Computer Science, vol. 11, p. e2921, 06 2025.

[20] T.-P. Chang, H.-M. Chen, S.-Y. Chen, and W.-C. Lin, "Deep learning model for dynamic hand gesture recognition for natural human-machine interface on end devices," International Journal of Information System Modeling and Design, vol. 13, no. 10, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1947818622000503

[21] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in CVPR 2011, 2011, pp. 1297–1304.

[22] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," Multimedia, IEEE Transactions on, vol. 15, pp. 1110–1120, 08 2013.

[23] C. Keskin, F. Kıraç, Y. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," 11 2011, pp. 1228–1234.

[24] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," 06 2015, pp. 1–7.

[25] O. Köpüklü, N. Köse, and G. Rigoll, "Motion fused frames: Data level fusion strategy for hand gesture recognition," 2018. [Online]. Available: https://arxiv.org/abs/1804.07187

[26] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," 2016. [Online]. Available: https://arxiv.org/abs/1607.07043

[27] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," Commun. ACM, vol. 54, no. 2, p. 60–71, Feb. 2011. [Online]. Available: https://doi.org/10.1145/1897816.1897838

[28] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in Computer Vision – ECCV 2012, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 872–885.

[29] Yaseen, "Real-time face gesture-based robot control using ghostnet in a unity simulation environment," Sensors, vol. 25, p. 6090, 10 2025.

[30] S. Kondratev, Y. Dyrchenkova, G. Nikitin, L. Voskov, V. Pikalov, and V. Meshcheryakov, "Aerokinesis: An iot-based vision-driven gesture control system for quadcopter navigation using deep learning and ros2," Technologies, vol. 14, no. 1, 2026. [Online]. Available: https://www.mdpi.com/2227-7080/14/1/69

[31] J. Wang, "Gesture based control technology for drones: Challenges and solutions," Applied and Computational Engineering, vol. 109, pp. 173–178, 11 2024.

[32] M. Obaid, F. Kistler, G. Kasparaviciute, E. Yantac, and M. Fjeld, "How would you gesture navigate a drone?: a user-centered approach to control a drone," 10 2016, pp. 113–121.

[33] J. Cauchard, K. Zhai, M. Spadafora, and J. Landay, "Emotion encoding in human-drone interaction," 03 2016, pp. 263–270.

[34] S. Benford, E. Schneiders, J. P. M. Avila, P. Caleb-Solly, P. R. Brundell, S. Castle-Green, F. Zhou, R. Garrett, K. Höök, S. Whatley, K. Marsh, and P. Tennent, "Somatic safety: An embodied approach towards safe human-robot interaction," 2025. [Online]. Available: https://arxiv.org/abs/2503.16960

[35] P. A. Lasota, T. Fong, and J. A. Shah, "A survey of methods for safe human-robot interaction," Foundations and Trends® in Robotics, vol. 5, no. 4, pp. 261–349, 2017. [Online]. Available: https://doi.org/10.1561/2300000052

[36] R. Tian, L. Sun, A. Bajcsy, M. Tomizuka, and A. D. Dragan, "Safety assurances for human-robot interaction via confidence-aware game-theoretic human models," in 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 11 229–11 235.

[37] M. Vierhauser, M. N. A. Islam, A. Agrawal, J. Cleland-Huang, and J. Mason, "Hazard analysis for human-on-the-loop interactions in suas systems," in ESEC/FSE '21: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, August 23-28, 2021, D. Spinellis, G. Gousios, M. Chechik, and M. D. Penta, Eds. ACM, 2021, pp. 8–19. [Online]. Available: https://doi.org/10.1145/3468264.3468534

[38] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (ears)," in 2009 17th IEEE International Requirements Engineering Conference, 2009, pp. 317–322.

[39] A. M. R. Bernal, W. J. Scheirer, and J. Cleland-Huang, "NOMAD: A natural, occluded, multi-scale aerial dataset, for emergency response scenarios," in IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024. IEEE, 2024, pp. 8569–8580. [Online]. Available: https://doi.org/10.1109/WACV57701.2024.00839

[40] A. M. R. Bernal, M. Petterson, P. A. Granadeno, M. Murphy, J. Mason, and J. Cleland-Huang, "Validating terrain models in digital twins for trustworthy suas operations," in 28th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2025 - Companion, Grand Rapids, MI, USA, October 5-10, 2025. IEEE, 2025, pp. 289–299. [Online]. Available: https://doi.org/10.1109/MODELS-C68889.2025.00047

[41] T. P. Chambers, P. Granadeno, U. Gohar, M. C. Hunter, A. M. R. Bernal, W. Tang, M. N. A. Islam, M. Cohen, T. Jung, R. Lutz, and J. Cleland-Huang, "Automated on-entry decision-making for utm zones based on reputations and certifications," in AIAA AVIATION FORUM AND ASCEND 2025, 2025, p. 3567.

[42] P. A. Granadeno, A. M. R. Bernal, M. N. A. Islam, and J. Cleland-Huang, "An environmentally complex requirement for safe separation distance between uavs," in 32nd IEEE International Requirements Engineering Conference, RE 2024 - Workshops, Reykjavik, Iceland, June 24-25, 2024. IEEE, 2024, pp. 166–175. [Online]. Available: https://doi.org/10.1109/REW61692.2024.00028

[43] J. Olesk, A. M. R. Bernal, and J. Cleland-Huang, "Momux: A catalyst for discovering situational awareness and team coordination requirements for human-autonomy teaming," in IEEE International Requirements Engineering Conference (RE), Montreal, Canada, 2026.