CG-XMARL v4.1 Technical Specification

Defense-Grade Multi-UAV Formation Pursuit Control System

Version: 4.1.0
Date: 2025-01-08

Author: Behnam Heydari

Executive Summary

CG-XMARL v4.1 is a military-grade multi-agent reinforcement learning system for autonomous UAV formation control integrated with DCS World 2.9. The system implements a 3-UAV wedge formation pursuit control architecture with advanced consensus-based virtual structure (CBVS) guidance, hierarchical finite state machines, and a planned HGAT-MAPPO-WM (Hierarchical Graph Attention Transformer with Multi-Agent Proximal Policy Optimization and World Model) neural architecture.

Key Capabilities

- **3-UAV Wedge Formation**: Rigid parallel heading formation with 1250m trail / 500m lateral offsets
- **Target Pursuit**: S-300 radar-guided target tracking with 6-10km optimal standoff
- **Triad Lifecycle Management**: Automatic replacement of entire UAV triads on loss/low-fuel
- **DCS World Integration**: Real-time gRPC communication with DCS World simulation
- **Threat Zone Avoidance**: WEZ-aware formation heading computation

Codebase Statistics

- **Total Python Files**: 709 (including generated protobuf, python, and wandb)
- **Core Training Modules**: 143 files in `training/` directory
- **Scripts**: 57 files in `scripts/` directory
- **DCS Interface**: 236 files in `dcs_interface/` directory

Teacher-Student Training Paradigm and Operational Scenario

Training Architecture Overview

This system employs an advanced **Teacher-Student training paradigm** where a sophisticated expert controller (AdvancedCBVSPursuitController v4.0) serves as the TEACHER to generate high-quality demonstration datasets before agent training begins. This approach enables:

1. **High-Quality Dataset Generation**: The Teacher controller generates demonstration trajectories across diverse conditions:
    - **Maps**: Multiple DCS World terrains (Caucasus, Nevada, Syria, Persian Gulf, etc.)
    - **Weather**: Variable conditions including clear, cloudy, rain, fog, and different wind patterns
    - **Positions**: Random spawn locations for blue flight and target aircraft
    - **Target Behavior**: Different flight paths, speeds, evasion patterns, and maneuvers
    - **Threat Zones**: Various WEZ configurations and placements
2. **World Model Training (Dreamer V3)**: The collected datasets train a world model that learns environment dynamics, enabling imagination-based training without expensive DCS simulation time.
3. **Imagination Pre-Training**: Agents undergo initial training in the World Model's imagination space using 15-step rollouts, dramatically reducing sample requirements.
4. **DCS Fine-Tuning**: Pre-trained agents are fine-tuned in the actual DCS World simulation for deployment readiness.

Current System Coverage

The implemented system covers the following capabilities:

1. **Wedge Formation Assembly**: Immediate formation of 3-UAV wedge upon takeoff (1250m trail, 500m lateral)
2. **Target Pursuit**: Tracking hostile aircraft at 6-10km optimal standoff distance
3. **Threat Zone Avoidance**: WEZ-aware heading computation to navigate around weapon engagement zones

Complete Operational Scenario

PHASE 1: STANDBY

- Three UAVs wait at origin airbase with engines OFF
- Aircraft are in INIT_PARKED state, Control Mode = STANDBY
- S-300 radar network continuously scans for hostile aircraft

PHASE 2: S-300 ACTIVATION

- When S-300 radars detect the target, an activation signal is generated
- Detection must persist for **100 consecutive steps** (anti-false-positive confirmation)
- Parameter: `s300_stability_steps = 100`
- Once confirmed: `s300_detection_confirmed = True`
- Aircraft transition: STANDBY → JOIN

PHASE 3: TAKEOFF AND FORMATION

- Engines start automatically upon activation
- Aircraft begin takeoff roll
- **CRITICAL**: Immediately upon becoming airborne, UAVs form wedge formation
- Teacher/CBVS controller guides formation assembly
- Formation geometry: Leader (Agent 0) at apex, Left Wing (Agent 1) at -1250m/-500m, Right Wing (Agent 2) at -1250m/+500m

PHASE 4: TARGET PURSUIT

- Formation tracks target at 6-10km optimal standoff
- Wedge formation maintained during pursuit maneuvers
- ATCFC (Adaptive Turn-Compensated Formation Control) ensures stability during turns
- Threat zones (WEZ) avoided through heading adjustments

PHASE 5: TARGET LOSS AND REACQUISITION (REACQ)

- If target exits S-300 AND onboard radar coverage during pursuit:
    - Aircraft enter REACQ control mode
    - Continue searching along last known heading
    - Formation maintained during search
- **Target Reappears**: System waits **120 consecutive steps** for confirmation (anti-false-positive), parameter adjustable
- **Timeout**: If target not reacquired within timeout period, aircraft enter RTB mode

PHASE 6: EPISODE TERMINATION

Episode ends when ANY of the following conditions occur:

Condition	Trigger	Action	Next State
Target Destroyed	Target unit destroyed	Immediate episode success	New episode
Target Parked	Target lands at any airbase	Episode success	New episode
Aircraft Destroyed	≥1 blue aircraft destroyed	Triad retirement → Reserve check	See below
Low Fuel	≥1 aircraft below 15% fuel	Triad retirement → Reserve check	See below
REACQ Timeout	Lost target permanently	Triad retirement → RTB	See below

TRIAD REPLACEMENT LOGIC:

When aircraft destruction, low fuel, or REACQ timeout occurs:

1. **Entire triad (all 3 aircraft) retires together** - group cohesion principle
2. Retiring aircraft enter RTB mode and return to base for landing
3. System checks reserve pool at origin airbase:
    - **≥3 reserves available**: New triad spawns, enters STANDBY, waits for fresh S-300 detection signal
    - **<3 reserves available**: Episode terminates (insufficient resources)

Key Parameters (Implemented Values):

Parameter	Value	Description
`s300_stability_steps`	100	S-300 detection confirmation steps
`reacq_timeout_s`	120.0s	REACQ search timeout
`fuel_threshold`	15%	Low fuel threshold triggering RTB
`d_trail`	1250m	Wedge trail distance
`d_lateral`	500m	Wedge lateral spacing
`d_optimal`	6-10km	Optimal pursuit standoff
`min_reserve_pool`	3	Minimum aircraft for triad replacement

Table of Contents

1. Formal Mathematical Foundations
2. Conceptual System Overview
3. Training Pipeline and Workflow
4. Operational Scenario
5. System Architecture Overview
6. Core Environment Layer
7. Guidance and Control Layer
8. Finite State Machine Layer
9. Perception and Observation Layer
10. Reward Architecture
11. Safety Architecture
12. DCS gRPC Interface
13. HGAT-MAPPO-WM Architecture
14. Data Flow and Pipeline
15. Configuration and Parameters
16. Appendices

1. Formal Mathematical Foundations

This section establishes the rigorous mathematical framework underlying CG-XMARL v4.1, providing formal definitions, theorems, and proofs that guarantee system stability and correctness.

1.1 Notation and Preliminaries

Definition 1.1 (State Space). The aircraft state space is defined as:

$$\xi_i = [p_i^T, v_i^T, \psi_i]^T \in \mathcal{X} \subset \mathbb{R}^7$$

where:

- $p_i = [n_i, e_i, h_i]^T \in \mathbb{R}^3$ is position in North-East-Down (NED) coordinates
- $v_i = [v_i^n, v_i^e, v_i^d]^T \in \mathbb{R}^3$ is velocity in NED frame
- $\psi_i \in [0, 2\pi)$ is heading angle

Definition 1.2 (Control Input Space). The control input space is:

$$u_i = [\psi_i^{cmd}, v_i^{cmd}, h_i^{cmd}]^T \in \mathcal{U} \subset \mathbb{R}^3$$

with bounds $\mathcal{U} = [\psi_{min}, \psi_{max}] \times [v_{min}, v_{max}] \times [h_{min}, h_{max}]$.

Definition 1.3 (Formation Graph). The formation topology is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where:

- $\mathcal{V} = \{0, 1, ..., N-1\}$ is the set of agents
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ defines communication links
- Agent 0 is designated the leader: $\nexists j : (j, 0) \in \mathcal{E}$

1.2 Wedge Formation Definition

Definition 1.4 (Wedge Formation Geometry). A wedge formation $\mathcal{W}$ for $N = 3$ agents is defined by the slot offset matrix:

$$\Delta = \begin{bmatrix} \delta_0 \\ \delta_1 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -d_{trail} & -d_{lat} \\ -d_{trail} & +d_{lat} \end{bmatrix} \in \mathbb{R}^{3 \times 2}$$

where $d_{trail} = 1250$ m and $d_{lat} = 500$ m are the trail and lateral offsets respectively.

Definition 1.5 (Slot Position). The world-frame slot position for agent i at time t is:

$$p_i^{slot}(t) = p_0(t) + R(\psi_L(t))\delta_i$$

where $R(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$ is the 2D rotation matrix and ψ_L is the formation heading.

Definition 1.6 (Formation Error). The formation error vector is:

$$\mathcal{F}(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} = \begin{bmatrix} p_1(t) - p_1^{slot}(t) \\ p_2(t) - p_2^{slot}(t) \end{bmatrix} \in \mathbb{R}^4$$

Definition 1.7 (Formation Quality Metric). Formation quality $Q \in [0, 1]$ is defined as:

$$Q(t) = 1 - \frac{1}{N-1} \sum_{i=1}^{N-1} \min\left(1, \frac{\|e_i(t)\|}{d_{threshold}}\right)$$

where $d_{threshold} = 500$ m. Perfect formation: $Q = 1$; complete divergence: $Q = 0$.

1.3 Consensus-Based Formation Control

Definition 1.8 (Consensus Protocol). The standard consensus protocol for formation control is:

$$\dot{p}_i = -\sum_{j \in \mathcal{N}_i} w_{ij}[(p_i - \delta_i) - (p_j - \delta_j)]$$

where $\mathcal{N}_i$ is the neighbor set of agent i and $w_{ij} > 0$ are edge weights.

Theorem 1.1 (Centroid Instability of Virtual Structure Methods). *Let $\bar{p}(t) = \frac{1}{N}\sum_{i=0}^{N-1} p_i(t)$ be the formation centroid. Under standard consensus with centroid-based reference, follower corrections cause $\frac{\partial \bar{p}}{\partial p_i} \neq 0$, creating a feedback loop that can lead to oscillation.*

Proof. Consider the centroid dynamics:

$$\dot{\bar{p}} = \frac{1}{N} \sum_{i=0}^{N-1} \dot{p}_i = \frac{1}{N} \sum_{i=0}^{N-1} u_i$$

When followers correct toward the centroid, their control inputs u_i depend on $\bar{p}$, which in turn depends on p_i. This creates the feedback:

$$\frac{d\bar{p}}{dt} = f(\bar{p}, \{p_i\}) \implies \frac{\partial \bar{p}}{\partial p_i} = \frac{1}{N} \neq 0$$

Under certain gain conditions, this feedback can exhibit limit cycle behavior. ∎

1.4 Fixed Leader Topology: Stability Guarantee

Definition 1.9 (Fixed Leader Protocol). The fixed leader formation protocol designates Agent 0 as permanent leader:

$$p_L(t) \triangleq p_0(t)$$

All slot positions reference this fixed point: $p_i^{slot}(t) = p_L(t) + R(\psi_L(t))\delta_i$.

Theorem 1.2 (Fixed Leader Centroid Independence). *Under the fixed leader protocol, the formation reference point is independent of follower positions:*

$$\frac{\partial p_L}{\partial p_i} = 0 \quad \forall i \neq 0$$

Proof. By definition, $p_L(t) = p_0(t)$. The leader's position evolves according to its pursuit dynamics:

$$\dot{p}_0 = f_{pursuit}(p_0, p_{target}, \mathcal{T})$$

where $\mathcal{T}$ represents threat zones. This is independent of $p_1, ..., p_{N-1}$. ∎

Corollary 1.1. *The fixed leader protocol eliminates centroid-induced oscillation.*

1.5 Rigid Parallel Heading Control Law

Control Law 1.1 (Leader Heading). The leader tracks the target:

$$\psi_0^{cmd}(t) = \arctan2(e_T - e_0, n_T - n_0)$$

Control Law 1.2 (Follower Parallel Heading). Followers fly parallel with small corrections:

$$\psi_i^{cmd}(t) = \psi_0(t) + k_{lat} \cdot e_i^{cross}(t)$$

where e_i^{cross} is the cross-track error:

$$e_i^{cross} = (p_i - p_i^{slot}) \cdot \hat{n}_{cross}, \quad \hat{n}_{cross} = [-\sin\psi_L, \cos\psi_L]^T$$

Theorem 1.3 (Exponential Convergence of Cross-Track Error). *Under Control Law 1.2 with $k_{lat} > 0$, the cross-track error converges exponentially:*

$$\|e_i^{cross}(t)\| \leq \|e_i^{cross}(0)\| \cdot e^{-k_{lat} v t}$$

Proof. Define Lyapunov function $V_i = \frac{1}{2}(e_i^{cross})^2$. Its time derivative:

$$\dot{V}_i = e_i^{cross} \cdot \dot{e}_i^{cross}$$

Under parallel heading with small lateral correction, the cross-track velocity is:

$$\dot{e}_i^{cross} \approx -k_{lat} \cdot v \cdot e_i^{cross}$$

Thus:

$$\dot{V}_i = -k_{lat} \cdot v \cdot (e_i^{cross})^2 = -2k_{lat} \cdot v \cdot V_i$$

By the comparison lemma: $V_i(t) \leq V_i(0)e^{-2k_{lat}vt}$

Taking square roots: $\|e_i^{cross}(t)\| \leq \|e_i^{cross}(0)\|e^{-k_{lat}vt}$ ∎

1.6 Control Barrier Functions for Safety

Definition 1.10 (Separation Barrier Function). For agents i, j, define:

$$h_{ij}(p_i, p_j) = \|p_i - p_j\|^2 - d_{safe}^2$$

where d_{safe} is the minimum separation distance.

Definition 1.11 (Safe Set). The safe set is:

$$\mathcal{C} = \{(p_0, ..., p_{N-1}) : h_{ij} \geq 0 \; \forall i \neq j\}$$

Theorem 1.4 (Forward Invariance via CBF). *If the control input satisfies:*

$$\dot{h}_{ij}(x, u) + \gamma h_{ij}(x) \geq 0 \quad \forall i \neq j$$

for some $\gamma > 0$, and $h_{ij}(x(0)) \geq 0$, then $h_{ij}(x(t)) \geq 0$ for all $t \geq 0$.

Proof. Standard CBF result (Ames et al., 2019). The condition ensures that when approaching the boundary of $\mathcal{C}$, the control steers the system away from violation. ∎

Implementation. The CBF condition is enforced via QP:

$$u^* = \arg\min_u \|u - u^{nom}\|^2$$
$$\text{s.t. } \nabla h_{ij}^T \cdot f(x, u) + \gamma h_{ij} \geq 0 \; \forall i \neq j$$

1.7 Adaptive Turn-Compensated Formation Control (ATCFC)

Definition 1.12 (Turn Rate). Formation turn rate:

$$\kappa(t) = \frac{d\psi_L}{dt}$$

estimated via exponential smoothing:

$$\hat{\kappa}(t) = \alpha \cdot \kappa_{raw}(t) + (1 - \alpha) \cdot \hat{\kappa}(t-1)$$

Proposition 1.1 (Turn-Induced Asymmetry). *During a turn with rate κ, an agent at cross-track offset d must travel arc length:*

$$s = (r \pm d) \cdot \theta = r\theta \pm d\theta$$

where r is the turn radius and θ is the turn angle. The asymmetry is $\pm d\theta = \pm d \cdot \kappa \cdot \Delta t$.

Control Law 1.3 (Turn Rate Feed-Forward - TRFF). The predictive heading correction:

$$\Delta\psi_{TRFF,i} = k_{trff} \cdot \hat{\kappa} \cdot \text{sign}(\delta_i^{cross})$$

Theorem 1.5 (Turn Lag Elimination). *With $k_{trff} = 1$, TRFF eliminates turn-induced lag for wing aircraft.*

Proof. The leader's heading rate is $\dot{\psi}_0 = \kappa$. Without TRFF, followers react with delay. With TRFF, each follower's commanded heading rate becomes:

$$\dot{\psi}_i^{cmd} = \dot{\psi}_0 + k_{trff} \cdot \kappa \cdot \text{sign}(\delta_i^{cross})$$

For parallel formation, $\text{sign}(\delta_i^{cross}) \cdot \kappa$ provides the anticipatory correction, making effective heading rate match the required rate for arc-length equality. ∎

Control Law 1.4 (Turn Speed Compensation). Geometric speed differential:

$$\Delta v_i = k_v \cdot \kappa \cdot \delta_i^{cross}$$

Inside wing ($\delta^{cross} < 0$, left turn): slower speed Outside wing ($\delta^{cross} > 0$, left turn): faster speed

1.8 Finite State Machine Formalization

Definition 1.13 (Deterministic FSM). A deterministic finite state machine is a 5-tuple:

$$M = (S, \Sigma, \delta, s_0, F)$$

where:

- S is a finite set of states
- Σ is the input alphabet
- $\delta : S \times \Sigma \to S$ is the transition function
- $s_0 \in S$ is the initial state
- $F \subseteq S$ is the set of final states

Definition 1.14 (Aircraft State FSM).

$$M_{aircraft} = (S_A, \Sigma_A, \delta_A, s_0^A, F_A)$$

where:

- $S_A = \{INIT\_PARKED, TAKEOFF, AIRBORNE, LOW\_FUEL, OP\_COMPLETE, PARKED, DESTROYED\}$
- $s_0^A = INIT\_PARKED$
- $F_A = \{PARKED, DESTROYED\}$

Definition 1.15 (Control Mode FSM).

$$M_{control} = (S_C, \Sigma_C, \delta_C, s_0^C, F_C)$$

where:

- $S_C = \{STANDBY, JOIN, PURSUIT, REACQ, RTB, INACTIVE\}$
- $s_0^C = STANDBY$
- $F_C = \{INACTIVE\}$

Theorem 1.6 (FSM Determinism). *Both $M_{aircraft}$ and $M_{control}$ are deterministic: for any state s and input σ, there exists exactly one transition.*

Proof. By construction, all transition guards are mutually exclusive and priority-ordered. ∎

Theorem 1.7 (FSM Termination). *The composed system $M_{aircraft} \times M_{control}$ always terminates in a valid final state.*

Proof. Every non-terminal state has at least one outgoing transition with eventually-satisfiable guard (timeout, fuel exhaustion, destruction). Terminal states have no outgoing transitions. By induction on mission time, the system reaches $F_A \times S_C$ or $S_A \times F_C$. ∎

1.9 Multi-Agent Reinforcement Learning Formulation

Definition 1.16 (Dec-POMDP). The formation control problem is formulated as a Decentralized Partially Observable Markov Decision Process:

$$\mathcal{M} = (\mathcal{S}, \{\mathcal{A}_i\}, \{\mathcal{O}_i\}, T, R, \{O_i\}, \gamma)$$

where:

- $\mathcal{S}$ is the global state space
- $\mathcal{A}_i$ is agent i's action space
- $\mathcal{O}_i$ is agent i's observation space
- $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function
- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function
- $O_i : \mathcal{S} \to \mathcal{O}_i$ is the observation function
- $\gamma \in [0, 1)$ is the discount factor

Definition 1.17 (CTDE Objective). Under Centralized Training with Decentralized Execution:

$$\max_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

subject to decentralized execution: $a_i^t \sim \pi_\theta(a_i | o_i^t)$.

Definition 1.18 (Counterfactual Advantage). For credit assignment:

$$\hat{A}_i^{CF}(s, a) = Q(s, a_i, a_{-i}) - \sum_{a_i'} \pi_i(a_i' | o_i) Q(s, a_i', a_{-i})$$

This isolates agent i's marginal contribution to the joint value.

1.10 Summary of Formal Guarantees

Property	Mathematical Guarantee	Reference
Centroid Stability	$\frac{\partial p_L}{\partial p_i} = 0$ for $i \neq 0$	Theorem 1.2
Cross-Track Convergence	Exponential: $O(e^{-k_{lat}vt})$	Theorem 1.3
Collision Avoidance	Forward invariance of $\mathcal{C}$	Theorem 1.4
Turn Lag Elimination	Zero lag with $k_{trff} = 1$	Theorem 1.5
FSM Determinism	Unique transitions	Theorem 1.6
FSM Termination	Guaranteed terminal state	Theorem 1.7

---

2. Conceptual System Overview

2.1 System Purpose and Philosophy

CG-XMARL v4.1 is designed to train autonomous multi-agent reinforcement learning (MARL) policies for 3-UAV wedge formation pursuit control in realistic military simulation environments. The system addresses the fundamental challenge of **coordinated multi-agent behavior** where agents must simultaneously:

1. **Maintain Rigid Formation**: Preserve geometric wedge structure during all maneuvers
2. **Pursue Dynamic Targets**: Track and follow hostile aircraft at optimal standoff distances
3. **Avoid Threat Zones**: Navigate around weapon engagement zones (WEZ) without breaking formation
4. **Handle Uncertainty**: Manage sensor noise, communication delays, and target loss scenarios

2.2 Core Operational Concept

```
┌─────────────────────────────────────────────────────────────────────┐
│                  CG-XMARL v4.1 OPERATIONAL CONCEPT                   │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│   ┌──────────┐  DETECT   ┌──────────┐  PURSUE  ┌──────────────────┐  │
│   │  S-300   │──────────►│   BLUE   │─────────►│      TARGET      │  │
│   │  RADAR   │  TARGET   │  FLIGHT  │   WITH   │ (Hostile Aircraft)│ │
│   │  NETWORK │           │ (3 UAVs) │ STANDOFF │                  │  │
│   └──────────┘           │ IN WEDGE │          │ ◄── Moving target│  │
│        │                 │ FORMATION│          │     that must be │  │
│        │                 └──────────┘          │     tracked      │  │
│        │                      │                └──────────────────┘  │
│        │                      │                                      │
│        │                 ┌──────────┐                                │
│        │                 │THREAT ZONES                               │
│        │                 │(WEZ Areas)│                               │
│        │                 │          │                                │
│        │                 │ ◄── Must be                               │
│        │                 │   avoided │                               │
│        │                 │   during  │                               │
│        │                 │   pursuit │                               │
│        │                 └──────────┘                                │
│        ▼                                                             │
│   ACTIVATION SIGNAL: When S-300 radars detect the target, they send an│
│   activation signal to the blue flight. Aircraft start engines, takeoff,│
│   form wedge formation, and begin pursuit.                           │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

2.3 Three Core Capabilities

Capability	Description	Implementation
Wedge Formation	3 aircraft maintain rigid V-shaped geometry with 500m trail / 600m lateral offsets	CBVS Controller with ATCFC
Target Pursuit	Track hostile aircraft at 6-10km optimal standoff distance	Sliding Mode Pursuit Control
Threat Avoidance	Route around weapon engagement zones without breaking formation	WEZ-aware heading computation

2.4 The Teacher-Student Paradigm

The system employs a **Teacher-Student** training paradigm where an expert controller (Teacher) generates high-quality demonstration data before training neural network agents (Students):

```
┌─────────────────────────────────────────────────────────────────────┐
│                        TEACHER-STUDENT PARADIGM                       │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  PHASE 1: DATA COLLECTION (Current Phase)                            │
│  ═══════════════════════════════════════                             │
│                                                                      │
│  ┌───────────────┐     ┌───────────────┐     ┌───────────────┐       │
│  │   TEACHER     │     │  DCS WORLD    │     │   DATASET     │       │
│  │  (CBVS v4.0)  │────►│  SIMULATION   │────►│ (Trajectories)│       │
│  │               │     │               │     │               │       │
│  │ Expert control│     │ Real physics  │     │ States, actions,      │
│  │ algorithms for│     │ and dynamics  │     │ rewards for each      │
│  │ formation and │     │ provide ground│     │ timestep across       │
│  │ pursuit       │     │ truth data    │     │ many episodes │       │
│  └───────────────┘     └───────────────┘     └───────────────┘       │
│                                                                      │
│  PHASE 2: WORLD MODEL TRAINING                                       │
│  ═════════════════════════════                                       │
│                                                                      │
│  ┌───────────────┐     ┌───────────────┐     ┌───────────────┐       │
│  │   DATASET     │     │  DREAMER V3   │     │  WORLD MODEL  │       │
│  │ (Trajectories)│────►│   TRAINING    │────►│(Learned Dynamics)     │
│  │               │     │               │     │               │       │
│  │ High-quality  │     │ Recurrent     │     │ Predicts next │       │
│  │ demonstration │     │ State-Space   │     │ states, rewards,      │
│  │ trajectories  │     │ Model (RSSM)  │     │ formations    │       │
│  └───────────────┘     └───────────────┘     └───────────────┘       │
│                                                                      │
│  PHASE 3: IMAGINATION PRE-TRAINING                                   │
│  ═════════════════════════════════                                   │
│                                                                      │
│  ┌───────────────┐     ┌───────────────┐     ┌───────────────┐       │
│  │  WORLD MODEL  │     │  HGAT-MAPPO   │     │  PRE-TRAINED  │       │
│  │ (Imagination) │────►│   TRAINING    │────►│    AGENTS     │       │
│  │               │     │               │     │               │       │
│  │ Generate      │     │ Train policies│     │ Agents learn  │       │
│  │ synthetic     │     │ in imagination│     │ formation and │       │
│  │ rollouts      │     │ (15-step)     │     │ pursuit skills│       │
│  └───────────────┘     └───────────────┘     └───────────────┘       │
│                                                                      │
│  PHASE 4: FINE-TUNING IN DCS                                         │
│  ═══════════════════════════                                         │
│                                                                      │
│  ┌───────────────┐     ┌───────────────┐     ┌───────────────┐       │
│  │  PRE-TRAINED  │     │  DCS WORLD    │     │DEPLOYMENT-READY       │
│  │    AGENTS     │────►│  FINE-TUNING  │────►│    AGENTS     │       │
│  │               │     │               │     │               │       │
│  │ Good initial  │     │ Adapt to real │     │ Robust policies       │
│  │ policies from │     │ simulation    │     │ for operational       │
│  │ imagination   │     │ dynamics      │     │ deployment    │       │
│  └───────────────┘     └───────────────┘     └───────────────┘       │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

---

2. Training Pipeline and Workflow

2.1 End-to-End Training Pipeline

```
┌─────────────────────────────────────────────────────────────────────┐
│                       COMPLETE TRAINING PIPELINE                      │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  STAGE 0: DATA COLLECTION                                            │
│  ════════════════════════                                            │
│                                                                      │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │ Script: collect_formation_data.py                             │   │
│  │ Controller: AdvancedCBVSPursuitController v4.0 (Teacher)      │   │
│  │                                                               │   │
│  │ For each episode:                                             │   │
│  │   1. Initialize DCS mission with varied conditions            │   │
│  │   2. Wait for S-300 detection activation                      │   │
│  │   3. Run Teacher controller to guide all aircraft             │   │
│  │   4. Record (observation, action, reward, next_obs) tuples    │   │
│  │   5. Save episode trajectory to HDF5/NPZ format               │   │
│  │                                                               │   │
│  │ Variation Dimensions:                                         │   │
│  │   • Maps: Different DCS terrains (Caucasus, Nevada, Syria, etc.)  │
│  │   • Weather: Clear, cloudy, rain, fog, wind conditions        │   │
│  │   • Positions: Random spawn locations for blue flight and target  │
│  │   • Target behavior: Different flight paths and speeds        │   │
│  │   • Threat zones: Various WEZ configurations                  │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│                              ▼                                       │
│  STAGE 1: WORLD MODEL PRE-TRAINING                                   │
│  ═════════════════════════════════                                   │
│                                                                      │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │ Location: training/marl/world_model/                          │   │
│  │ Architecture: Dreamer V3 with Formation-Aware Decoders        │   │
│  │                                                               │   │
│  │ Components:                                                   │   │
│  │   • RSSM: Recurrent State-Space Model for dynamics            │   │
│  │   • Encoder: Observation → Latent state                       │   │
│  │   • Dynamics: Predict h_{t+1} from (h_t, z_t, a_t)            │   │
│  │   • Formation Decoder: Predict slot errors, quality metrics   │   │
│  │   • Reward Predictor: Predict r_t from latent state           │   │
│  │                                                               │   │
│  │ Training Objective:                                           │   │
│  │   L = L_reconstruction + L_dynamics + L_reward + L_KL         │   │
│  │                                                               │   │
│  │ Success Criteria: Reconstruction MSE < 0.01                   │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│                              ▼                                       │
│  STAGE 2: IMAGINATION TRAINING (HGAT-MAPPO)                          │
│  ══════════════════════════════════════════                          │
│                                                                      │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │ Location: training/marl/hgat_mappo/                           │   │
│  │ Architecture: Hierarchical Graph Attention + MAPPO            │   │
│  │                                                               │   │
│  │ Imagination Rollouts:                                         │   │
│  │   1. Sample initial state from replay buffer                  │   │
│  │   2. Use World Model to generate 15-step imagined trajectories│   │
│  │   3. Train HGAT-MAPPO on imagined data (no DCS simulation needed) │
│  │   4. Repeat for millions of imagined steps                    │   │
│  │                                                               │   │
│  │ Key Benefit:                                                  │   │
│  │   • 1000x faster than real DCS simulation                     │   │
│  │   • Explore diverse scenarios without waiting for DCS         │   │
│  │   • Safe exploration of dangerous maneuvers                   │   │
│  │                                                               │   │
│  │ Success Criteria: Formation quality > 0.95 in imagination     │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│                              ▼                                       │
│  STAGE 3: DCS FINE-TUNING                                            │
│  ════════════════════════                                            │
│                                                                      │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │ Location: training/rl/trainer.py                              │   │
│  │ Environment: Live DCS World 2.9 simulation                    │   │
│  │                                                               │   │
│  │ Fine-tuning Process:                                          │   │
│  │   1. Load pre-trained agents from imagination training        │   │
│  │   2. Run in DCS with reduced learning rate                    │   │
│  │   3. Adapt to real simulation dynamics (sim-to-real transfer) │   │
│  │   4. Validate performance metrics                             │   │
│  │                                                               │   │
│  │ Success Criteria:                                             │   │
│  │   • Formation quality > 0.80 in real DCS                      │   │
│  │   • Pursuit success > 90%                                     │   │
│  │   • Robust to 20 m/s crosswind                                │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

2.2 Teacher Controller: CBVS v4.0

The Teacher controller (`AdvancedCBVSPursuitController`) is a handcrafted expert system that generates high-quality demonstration data:

Algorithm	Purpose	Key Feature
CBVS	Consensus-Based Virtual Structure	Maintains formation geometry
ATCFC	Adaptive Turn-Compensated Formation Control	Handles turns without breaking formation
TRFF	Turn Rate Feed-Forward	Wings turn simultaneously with leader
CBF	Control Barrier Function	Safety guarantees during maneuvers
Hungarian	Optimal Slot Assignment	Stable wing-to-slot mapping

2.3 World Model: Dreamer V3 Architecture

```
┌─────────────────────────────────────────────────────────────────┐
│                   DREAMER V3 WORLD MODEL ARCHITECTURE             │
├─────────────────────────────────────────────────────────────────┤
│                                                                  │
│  Observation (o_t)                                               │
│       │                                                          │
│       ▼                                                          │
│  ┌──────────┐   CNN + MLP                                        │
│  │ Encoder  │   Maps observation to stochastic latent            │
│  │ q(z|h,o) │                                                    │
│  └──────────┘                                                    │
│       │ z_t (stochastic latent)                                  │
│       │                                                          │
│       ▼                                                          │
│  ┌───────────────────────────────────────────────────────────┐   │
│  │                          RSSM                             │   │
│  │                                                           │   │
│  │  ┌──────────┐     ┌──────────┐      ┌──────────┐          │   │
│  │  │ Sequence │     │ Dynamics │      │ Posterior│          │   │
│  │  │ Model    │────►│ Predictor│─────►│ (for z_t)│          │   │
│  │  │ h_t = f(.)│    │ p(z|h)   │      │ q(z|h,o) │          │   │
│  │  └──────────┘     └──────────┘      └──────────┘          │   │
│  │                                                           │   │
│  │  h_t = GRU(h_{t-1}, z_{t-1}, a_{t-1})                     │   │
│  └───────────────────────────────────────────────────────────┘   │
│       │                                                          │
│       │ (h_t, z_t) = Model State                                 │
│       │                                                          │
│       ├──────────────┬──────────────┬──────────────┐             │
│       ▼              ▼              ▼              ▼             │
│  ┌──────────┐  ┌──────────┐  ┌──────────┐  ┌───────────────┐     │
│  │ Obs      │  │ Reward   │  │ Continue │  │ Formation     │     │
│  │ Decoder  │  │ Predictor│  │ Predictor│  │ Decoder       │     │
│  │ p(o|h,z) │  │ p(r|h,z) │  │ p(c|h,z) │  │ Slot errors, etc.   │
│  └──────────┘  └──────────┘  └──────────┘  └───────────────┘     │
│                                                                  │
└─────────────────────────────────────────────────────────────────┘
```

---

3. Operational Scenario

3.1 Mission Lifecycle Overview

```
┌─────────────────────────────────────────────────────────────────────┐
│                        COMPLETE MISSION LIFECYCLE                     │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │ PHASE 1: STANDBY (Engines Off, Awaiting Activation)           │   │
│  ├───────────────────────────────────────────────────────────────┤   │
│  │                                                               │   │
│  │ Aircraft State: INIT_PARKED                                   │   │
│  │ Control Mode: STANDBY                                         │   │
│  │                                                               │   │
│  │ Blue Flight (3 UAVs) parked at origin airbase with engines off.   │
│  │ S-300 radar network continuously scanning for hostile aircraft.   │
│  │                                                               │   │
│  │  ┌────────┐                             ┌────────┐            │   │
│  │  │ Blue-1 │                             │ S-300  │            │   │
│  │  │ Blue-2 │   WAITING                   │ RADAR  │ ◄── Scanning   │
│  │  │ Blue-3 │                             │        │            │   │
│  │  └────────┘                             └────────┘            │   │
│  │                                                               │   │
│  │ TRIGGER: S-300 detects target → Activation signal sent        │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│                              ▼                                       │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │ PHASE 2: TAKEOFF AND FORMATION (Forming Wedge)                │   │
│  ├───────────────────────────────────────────────────────────────┤   │
│  │                                                               │   │
│  │ Aircraft State: TAKEOFF → AIRBORNE                            │   │
│  │ Control Mode: JOIN                                            │   │
│  │                                                               │   │
│  │ 1. Engines start automatically                                │   │
│  │ 2. Aircraft begin takeoff roll                                │   │
│  │ 3. Once airborne, begin forming wedge formation               │   │
│  │ 4. Teacher/CBVS controller guides formation assembly          │   │
│  │                                                               │   │
│  │                    ▲ To Target                                │   │
│  │                    │                                          │   │
│  │            ┌───────●───────┐ LEADER                           │   │
│  │            │               │                                  │   │
│  │         ◄──┘               └──►                               │   │
│  │    LEFT_WING              RIGHT_WING                          │   │
│  │                                                               │   │
│  │ TRIGGER: Formation quality > threshold → Enter PURSUIT        │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│                              ▼                                       │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │ PHASE 3: PURSUIT (Tracking Target with Standoff)              │   │
│  ├───────────────────────────────────────────────────────────────┤   │
│  │                                                               │   │
│  │ Aircraft State: AIRBORNE                                      │   │
│  │ Control Mode: PURSUIT                                         │   │
│  │                                                               │   │
│  │ • Formation tracks target at 6-10km optimal standoff          │   │
│  │ • Wedge formation maintained during pursuit                   │   │
│  │ • Threat zones (WEZ) avoided through heading adjustments      │   │
│  │ • ATCFC ensures formation stability during turns              │   │
│  │                                                               │   │
│  │  ┌───────────┐                     ┌────────┐                 │   │
│  │  │Wedge Flight│ ──────────────────►│ TARGET │                 │   │
│  │  │(Formation)│      6-10km         │        │                 │   │
│  │  │           │      standoff       │        │                 │   │
│  │  └───────────┘                     └────────┘                 │   │
│  │                         ▲                                     │   │
│  │                         │                                     │   │
│  │                  ┌─────────────┐                              │   │
│  │                  │ Threat Zone │ ◄── Avoided                  │   │
│  │                  │    (WEZ)    │                              │   │
│  │                  └─────────────┘                              │   │
│  │                                                               │   │
│  │ POSSIBLE TRANSITIONS:                                         │   │
│  │   • Target lost → REACQ mode                                  │   │
│  │   • Low fuel → RTB mode                                       │   │
│  │   • Aircraft destroyed → Triad retirement                     │   │
│  │   • Target destroyed → Episode success                        │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

3.2 Target Loss and Reacquisition (REACQ Mode)

When the target exits S-300 and onboard radar coverage during pursuit:

```
┌─────────────────────────────────────────────────────────────────────┐
│                      REACQUISITION (REACQ) PROTOCOL                   │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  TRIGGER: Target disappears from both S-300 and onboard radars       │
│                                                                      │
│  1. ENTER REACQ MODE                                                 │
│     └─► Control Mode: PURSUIT → REACQ                                │
│     └─► Start timeout counter                                        │
│                                                                      │
│  2. SEARCH PATTERN                                                   │
│     └─► Aircraft continue last known heading                         │
│     └─► Formation maintained during search                           │
│                                                                      │
│  3. IF TARGET REAPPEARS:                                             │
│     └─► Wait 120 consecutive steps for confirmation (anti-false-positive)│
│     └─► If confirmed: REACQ → PURSUIT (resume tracking)              │
│                                                                      │
│  4. IF TIMEOUT REACHED:                                              │
│     └─► REACQ → RTB (return to base)                                 │
│     └─► Triggers triad retirement                                    │
│                                                                      │
│  ┌──────────┐                                                        │
│  │ Timeline │                                                        │
│  ├──────────┴─────────────────────────────────────────────────────┐  │
│  │                                                                │  │
│  │  Target Lost   Search Period    Target Found?    Confirmation  │  │
│  │      │              │                │                │        │  │
│  │      ▼              ▼                ▼                ▼        │  │
│  │  ────●──────────────────────────────●────────────────●──────►  │  │
│  │      │                              │                │         │  │
│  │   Enter REACQ                   Reacquire      Wait 120 steps  │  │
│  │                                                then PURSUIT    │  │
│  │                                                                │  │
│  │  OR if timeout:                                                │  │
│  │  ────●───────────────────────────────────────────────●──────►  │  │
│  │      │                                               │         │  │
│  │   Enter REACQ                                   Timeout → RTB  │  │
│  │                                                                │  │
│  └────────────────────────────────────────────────────────────────┘  │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

3.3 Episode Termination Conditions

An episode ends when ANY of the following conditions are met:

Condition	Type	Result	Next Action
Target Destroyed	Success	Episode complete	New episode
Target Parked	Success	Target landed at airbase	New episode
Aircraft Destroyed	Failure	≥1 aircraft lost	Triad replacement OR episode end
Low Fuel	Neutral	≥1 aircraft below 15%	Triad replacement OR episode end
REACQ Timeout	Failure	Lost target permanently	Triad retirement → RTB
No Reserves	Terminal	<3 aircraft at airbase	Episode end

3.4 Triad Replacement Logic

```
┌─────────────────────────────────────────────────────────────────────┐
│                         TRIAD REPLACEMENT LOGIC                       │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  TRIGGER: Aircraft destroyed OR low fuel OR REACQ timeout            │
│                                                                      │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │                         DECISION TREE                         │   │
│  ├───────────────────────────────────────────────────────────────┤   │
│  │                                                               │   │
│  │                  Triad Retirement Triggered                   │   │
│  │                              │                                │   │
│  │                              ▼                                │   │
│  │                 ┌────────────────────────┐                    │   │
│  │                 │ Check Available Reserves│                   │   │
│  │                 │ at Origin Airbase      │                    │   │
│  │                 └────────────────────────┘                    │   │
│  │                              │                                │   │
│  │            ┌─────────────────┼─────────────────┐              │   │
│  │            │                 │                 │              │   │
│  │       ≥3 aircraft      1-2 aircraft       0 aircraft          │   │
│  │            │                 │                 │              │   │
│  │            ▼                 ▼                 ▼              │   │
│  │  ┌────────────────┐  ┌──────────┐  ┌──────────────┐           │   │
│  │  │ SPAWN NEW TRIAD│  │ EPISODE  │  │   EPISODE    │           │   │
│  │  │                │  │   END    │  │     END      │           │   │
│  │  │ 1. Retire old  │  │          │  │              │           │   │
│  │  │    flight (RTB)│  │ Not      │  │ No reserves  │           │   │
│  │  │                │  │ enough   │  │ available    │           │   │
│  │  │ 2. Assign 3 new│  │ reserves │  │              │           │   │
│  │  │    aircraft    │  └──────────┘  └──────────────┘           │   │
│  │  │                │                                           │   │
│  │  │ 3. New triad   │                                           │   │
│  │  │    enters      │                                           │   │
│  │  │    STANDBY     │                                           │   │
│  │  │                │                                           │   │
│  │  │ 4. Wait for    │                                           │   │
│  │  │    S-300 signal│                                           │   │
│  │  │                │                                           │   │
│  │  │ 5. Continue    │                                           │   │
│  │  │    mission     │                                           │   │
│  │  └────────────────┘                                           │   │
│  │                                                               │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                                                                      │
│  KEY BEHAVIOR:                                                       │
│  • Entire triad retires together (group cohesion)                    │
│  • New triad waits for fresh S-300 detection signal                  │
│  • S-300 detection state preserved (s300_detection_confirmed=True)   │
│  • Seamless handoff between triads                                   │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

3.5 S-300 Detection and Activation Signal

```
┌─────────────────────────────────────────────────────────────────────┐
│                       S-300 DETECTION AND ACTIVATION                  │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  The S-300 radar network serves as the PRIMARY ACTIVATION TRIGGER:   │
│                                                                      │
│  1. DETECTION PHASE                                                  │
│     └─► S-300 radars scan for hostile aircraft                       │
│     └─► Detection event triggers stability counter                   │
│                                                                      │
│  2. CONFIRMATION PHASE (Anti-False-Positive)                         │
│     └─► Detection must persist for 100 consecutive steps             │
│     └─► Parameter: s300_stability_steps = 100                        │
│     └─► Prevents activation on transient radar returns               │
│                                                                      │
│  3. ACTIVATION SIGNAL                                                │
│     └─► Once confirmed, s300_detection_confirmed = True              │
│     └─► Blue flight transitions: STANDBY → JOIN                      │
│     └─► Engines start, takeoff begins                                │
│                                                                      │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │                       DETECTION TIMELINE                      │   │
│  ├───────────────────────────────────────────────────────────────┤   │
│  │                                                               │   │
│  │  Target enters    Stability check    Activation    Blue flight│   │
│  │  S-300 range      (100 steps)        confirmed     takes off  │   │
│  │       │                │                 │              │     │   │
│  │       ▼                ▼                 ▼              ▼     │   │
│  │  ─────●────────────────────────────────●──────────────●────►  │   │
│  │       │                                │              │       │   │
│  │  detection_ok=True          s300_detection_confirmed   JOIN   │   │
│  │  s300_detected_consecutive=0     =True                 mode   │   │
│  │       │                                                       │   │
│  │       └──────► Counter increments each step until 100         │   │
│  │                                                               │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

3.6 Complete State Machine Integration

```
┌─────────────────────────────────────────────────────────────────────┐
│                      INTEGRATED FSM STATE TRANSITIONS                 │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  AIRCRAFT STATE FSM            CONTROL MODE FSM                      │
│  ══════════════════            ════════════════                      │
│                                                                      │
│  ┌─────────────┐              ┌─────────────┐                        │
│  │ INIT_PARKED │◄────────────►│   STANDBY   │  Engines off, awaiting │
│  └─────────────┘              └─────────────┘                        │
│         │                            │                               │
│         │ S-300 detection            │ S-300 detection               │
│         ▼                            ▼                               │
│  ┌─────────────┐              ┌─────────────┐                        │
│  │   TAKEOFF   │◄────────────►│    JOIN     │  Forming wedge         │
│  └─────────────┘              └─────────────┘                        │
│         │                            │                               │
│         │ inAir flag                 │ Formation assembled           │
│         ▼                            ▼                               │
│  ┌─────────────┐              ┌─────────────┐                        │
│  │  AIRBORNE   │◄────────────►│   PURSUIT   │  Tracking target       │
│  └─────────────┘              └─────────────┘                        │
│         │                            │                               │
│         │                            │ Target lost                   │
│         │                            ▼                               │
│         │                     ┌─────────────┐                        │
│         │                     │    REACQ    │  Searching for target  │
│         │                     └─────────────┘                        │
│         │                            │                               │
│         │ Low fuel / destroyed       │ Timeout / low fuel            │
│         ▼                            ▼                               │
│  ┌─────────────┐              ┌─────────────┐                        │
│  │  LOW_FUEL   │◄────────────►│     RTB     │  Return to base        │
│  └─────────────┘              └─────────────┘                        │
│         │                            │                               │
│         │ Landed                     │ Landed                        │
│         ▼                            ▼                               │
│  ┌─────────────┐              ┌─────────────┐                        │
│  │   PARKED    │◄────────────►│  INACTIVE   │  Terminal state        │
│  └─────────────┘              └─────────────┘                        │
│                                                                      │
│  ┌─────────────┐                                                     │
│  │  DESTROYED  │ ◄── Any state can transition here on unit death     │
│  └─────────────┘                                                     │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

---

4. System Architecture Overview

4.1 Layered Architecture Diagram

```
┌─────────────────────────────────────────────────────────────────────┐
│                        CG-XMARL v4.1 SYSTEM ARCHITECTURE              │
├─────────────────────────────────────────────────────────────────────┤
│                                                                      │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │                  LAYER 1: GYM/GYMNASIUM INTERFACE             │   │
│  │ DCSMultiAgentEnv ─ OpenAI Gym-compatible wrapper              │   │
│  │ • reset() / step() / render() / close()                       │   │
│  │ • Observation Space: Box(48,) per agent                       │   │
│  │ • Action Space: Box(3,) [heading_delta, speed_delta, altitude_delta]│
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │                  LAYER 2: MODULAR PROCESSING                  │   │
│  │ ┌────────────┐ ┌────────────┐ ┌────────────┐ ┌────────────┐   │   │
│  │ │ Perception │ │  Commands  │ │  Rewards   │ │  Threats   │   │   │
│  │ │ dcs_percep │ │ dcs_command│ │ dcs_rewards│ │ dcs_threats│   │   │
│  │ │ ────────── │ │ ────────── │ │ ────────── │ │ ────────── │   │   │
│  │ │ 48-dim obs │ │HSA pipeline│ │Hierarchical│ │WEZ computation │   │
│  │ │construction│ │ 5-stage    │ │form+pursuit│ │Zone avoidance  │   │
│  │ └────────────┘ └────────────┘ └────────────┘ └────────────┘   │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │                   LAYER 3: SAFETY ARCHITECTURE                │   │
│  │ ┌────────────────────────┐ ┌────────────────────────────────┐ │   │
│  │ │     Safety Shield      │ │        Cohesion Filter         │ │   │
│  │ │ • Envelope protection  │ │ • Origin-based grouping        │ │   │
│  │ │ • Rate limiting        │ │ • Soft/hard radius bounds      │ │   │
│  │ │ • Fail-safe recovery   │ │ • Geofence enforcement         │ │   │
│  │ └────────────────────────┘ └────────────────────────────────┘ │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │                   LAYER 4: FSM-BASED LIFECYCLE                │   │
│  │ ┌────────────────────────┐ ┌────────────────────────────────┐ │   │
│  │ │   Aircraft State FSM   │ │       Control Mode FSM         │ │   │
│  │ │ INIT_PARKED → TAKEOFF →│ │ STANDBY → JOIN → PURSUIT →     │ │   │
│  │ │ AIRBORNE → LOW_FUEL/   │ │ REACQ → RTB → INACTIVE         │ │   │
│  │ │ OP_COMPLETE → PARKED   │ │                                │ │   │
│  │ └────────────────────────┘ └────────────────────────────────┘ │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │                   LAYER 5: DCS-gRPC INTERFACE                 │   │
│  │ DCSGrpcClient ─ Military-grade gRPC wrapper                   │   │
│  │ • Connection lifecycle with auto-reconnect                    │   │
│  │ • Batched HSA command transmission                            │   │
│  │ • State queries (transforms, AGL, fuel, radar)                │   │
│  │ • Lua code generation and execution                           │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                              │                                       │
│  ┌───────────────────────────────────────────────────────────────┐   │
│  │                   LAYER 6: GUIDANCE CONTROLLERS               │   │
│  │ ┌───────────────────────────────────────────────────────────┐ │   │
│  │ │             AdvancedCBVSPursuitController v4.0            │ │   │
│  │ │ • Consensus-Based Virtual Structure (CBVS) formation      │ │   │
│  │ │ • Sliding Mode Pursuit Control                            │ │   │
│  │ │ • Control Barrier Function (CBF) safety                   │ │   │
│  │ │ • Hungarian Algorithm slot assignment                     │ │   │
│  │ │ • Adaptive Turn-Compensated Formation Control (ATCFC)     │ │   │
│  │ │ • Turn Rate Feed-Forward (TRFF)                           │ │   │
│  │ └───────────────────────────────────────────────────────────┘ │   │
│  └───────────────────────────────────────────────────────────────┘   │
│                                                                      │
└─────────────────────────────────────────────────────────────────────┘
```

5.2 Key Components Summary

Component	File	Purpose
Core Environment	`training/environments/dcs_multi_agent_env.py`	Gym wrapper, state management, lifecycle
CBVS Controller	`training/guidance/formation_cbvs_v2.py`	Formation control, pursuit guidance
Perception	`training/environments/dcs_perception.py`	48-dim observation construction
Commands	`training/environments/dcs_commands.py`	HSA command pipeline
Rewards	`training/environments/dcs_rewards.py`	Hierarchical reward computation
Safety Shield	`training/safety/safety_shield.py`	HSA filtering, envelope protection
Aircraft FSM	`training/fsm/aircraft_state_fsm.py`	Aircraft lifecycle states
Control FSM	`training/fsm/control_mode_fsm.py`	Tactical control modes
gRPC Client	`dcs_interface/grpc/client.py`	DCS World communication

---

5. Core Environment Layer

5.1 DCSMultiAgentEnv Class

Location: `training/environments/dcs_multi_agent_env.py`

The `DCSMultiAgentEnv` class serves as the OpenAI Gym/Gymnasium wrapper for the 3-UAV multi-agent formation pursuit control system.

5.1.1 EnvState Dataclass

Centralized container for all mutable environment state:

```python
@dataclass
class EnvState:
    # Connection
    client: Optional[DCSGrpcClient] = None
    connected: bool = False

    # Step tracking
    step_count: int = 0
    episode_count: int = 0

    # Detection
    detection_ok: bool = False
    s300_detection_confirmed: bool = False
    s300_detected_consecutive_steps: int = 0

    # Triad management
    triad_enable: bool = True
    triad_retire_active: bool = False
    triad_replacement_pending: bool = False
    replacement_units: Set[str] = field(default_factory=set)

    # Unit tracking
    active_unit_by_aid: Dict[str, str] = field(default_factory=dict)
    available_reserve: List[str] = field(default_factory=list)
    ignored_units_by_name: Set[str] = field(default_factory=set)

    # Wedge formation
    wedge_lock_released: bool = False
    wedge_relocked: bool = False

    # Threat zones
    threat_zones: List[Dict[str, Any]] = field(default_factory=list)
```

5.1.2 Observation Space

48-dimensional observation vector per agent:

Layer	Dimensions	Features
Sensor Features	12	yaw sin/cos, speed, altitude, vspeed, target dist/bearing, closure rate, radar/datalink flags
Own State	12	Position NED (norm), Velocity NED (norm), Attitude, Angular rates
Wingmen Relative	16	2 × (rel_pos 4, rel_vel 2, range, rel_speed)
Teammate Speeds	2	Normalized speeds for coordination
Target Block	6	Target rel_pos 3, rel_vel 3

5.1.3 Action Space

3-dimensional continuous action per agent:

Index	Action	Range	Scale
0	Heading Delta	[-1, 1]	±30° default
1	Speed Delta	[-1, 1]	±50 m/s default
2	Altitude Delta	[-1, 1]	±500m default

5.2 Mission Reset Modes

The environment supports multiple reset modes for training:

Mode	Method	Use Case
`none`	No reset	Continuous operation
`reload_mission`	`net.load_mission`	Dedicated server
`trigger_load`	Mission trigger flag	Single-player
`webgui_selenium`	Selenium automation	WebGUI server

---

6. Guidance and Control Layer

6.1 AdvancedCBVSPursuitController v4.0

Location: `training/guidance/formation_cbvs_v2.py`

Military-grade hybrid controller combining multiple control algorithms.

6.1.1 Theoretical Foundation

```
RIGID PARALLEL HEADING FORMATION v2.0

All aircraft:
1. Fly PARALLEL headings (same direction as leader)
2. Share SYNCHRONIZED speed (no rubber-band effect)
3. Maintain fixed geometric offsets through small corrections

MATHEMATICAL FORMULATION
========================

Leader target heading:
    leader_target_heading = atan2(target_e - leader_e, target_n - leader_n)

Shared speed:
    shared_speed = clamp(mean(all_speeds), v_min, v_max)

Per-aircraft commands:
    LEADER (Agent 0):
        heading_cmd = (leader_target_heading - current_heading) / π × 2
        speed_cmd = (shared_speed - current_speed) / 50

    WINGS (Agents 1, 2):
        # Parallel heading (dominant)
        parallel_cmd = (leader_target_heading - current_heading) / π × 2

        # Small lateral correction
        e_cross = cross_track_error(agent_pos, slot_pos, formation_heading)
        lateral_cmd = clamp(e_cross × k_lateral, -0.3, 0.3)

        heading_cmd = parallel_cmd + 0.2 × lateral_cmd
```

6.1.2 Wedge Formation Geometry

```
                  ▲ Formation Heading
                  │
                  │
          ┌───────●───────┐  LEADER (Agent 0)
          │       │       │  offset: (0, 0)
          │       │       │
  -600m ◄─┼───────┼───────┼─► +600m  (cross-track)
          │       │       │
          │       │       │
     ◄────┘       │       └────►
  LEFT_WING       │        RIGHT_WING
  (Agent 1)       │        (Agent 2)
  (-500, -600)    │        (-500, +600)
                  │
                  ▼ -500m (along-track / trail)
```

6.1.3 Control Phases

Phase	Distance	Formation Weight	Pursuit Weight
FORMING	Initial	1.0	0.0
INTERCEPT	>25km	0.3	0.7
APPROACH	10-25km	0.5	0.5
TRACKING	5-10km	0.7	0.3
STATION	<5km	0.8	0.2

6.1.4 Adaptive Turn-Compensated Formation Control (ATCFC)

During turns, the controller:

1. **Detects Turn Rate**: Monitors formation heading changes
2. **Computes Side Bias**: Keeps wings on correct side during turns
3. **Turn Rate Feed-Forward (TRFF)**: Wings turn simultaneously with leader
4. **Geometric Speed Compensation**: Inside wing slows, outside wing speeds up

```python
# Turn speed compensation
turn_dir = 1.0 if turn_rate >= 0 else -1.0  # +1 = right turn
wing_side = 1.0 if nominal_cross >= 0 else -1.0  # +1 = right wing
is_inside = turn_dir * wing_side > 0

if is_inside:
    turn_speed_adj = -geometric_delta_v  # Slow down
else:
    turn_speed_adj = +geometric_delta_v  # Speed up
```

6.1.5 WEZ Avoidance

Formation heading is computed with threat zone awareness:

```python
formation_heading = compute_join_heading_wez_aware(
    current_uv=leader_uv,
    join_target_uv=target_uv,
    zones=threat_zones,
    margin_m=wez_margin_m,
    turn_buffer_m=wez_turn_buffer_m,
)
```

---

7. Finite State Machine Layer

7.1 Aircraft State FSM

Location: `training/fsm/aircraft_state_fsm.py`

```
┌─────────────────────────────────────────────────────────────┐
│                     AIRCRAFT LIFECYCLE FSM                   │
├─────────────────────────────────────────────────────────────┤
│                                                             │
│   [*] ──────────────────────────────────────┐               │
│                                             ▼               │
│                                     ┌──────────────┐        │
│                                     │ INIT_PARKED  │        │
│                                     │  (initial)   │        │
│                                     └──────────────┘        │
│                                             │               │
│                                        [S1] │ JOIN cmd      │
│                                             ▼               │
│                                     ┌──────────────┐        │
│                                     │   TAKEOFF    │        │
│                                     │  (rolling)   │        │
│                                     └──────────────┘        │
│                                             │               │
│                                        [S2] │ inAir         │
│                                             ▼               │
│       ┌──────────────┐       [S3]   ┌──────────────┐        │
│       │   LOW_FUEL   │◄──────────── │   AIRBORNE   │        │
│       │ (emergency)  │   fuel<15%   │   (normal)   │        │
│       └──────────────┘              └──────────────┘        │
│              │                              │               │
│         [S5] │                         [S4] │ success       │
│              │                              ▼               │
│              │                      ┌──────────────┐        │
│              │                      │ OP_COMPLETE  │        │
│              │                      │  (success)   │        │
│              │                      └──────────────┘        │
│              │                              │               │
│              └──────────────┬───────────────┘               │
│                             ▼                               │
│                     ┌──────────────┐                        │
│                     │    PARKED    │                        │
│                     │  (terminal)  │                        │
│                     └──────────────┘                        │
│                                                             │
│      (any state) ──► ┌──────────────┐ ◄── unit death        │
│                      │  DESTROYED   │                       │
│                      │  (terminal)  │                       │
│                      └──────────────┘                       │
│                                                             │
└─────────────────────────────────────────────────────────────┘
```

7.2 Control Mode FSM

Location: `training/fsm/control_mode_fsm.py`

Mode	Description	Command Authority
STANDBY	On Ground, Awaiting	No commands (engines off)
JOIN	Formation Phase	Teacher/Autopilot
PURSUIT	Active Target Tracking	RL Policy / Teacher
REACQ	Target Lost - Searching	Search Pattern Controller
RTB	Return to Base	DCS AI Autopilot
INACTIVE	Mission Complete	No commands (terminal)

Transition Priority Order (Highest to Lowest):

1. P0: INACTIVE terminal check
2. P1: RTB enforcement (one-way)
3. P2: RTB triggers (fuel/emergency/mission-complete)
4. P3: JOIN (S-300 detection)
5. P4: REACQ (target loss recovery)
6. P5: PURSUIT (target detected)
7. P6: STANDBY (ground-only fallback)

---

8. Perception and Observation Layer

8.1 Observation Construction

Location: `training/environments/dcs_perception.py`

```python
def compute_sensor_features(env, tr, target_tr, ...) -> np.ndarray:
    """
    Output indices:
        [0]: yaw_sin
        [1]: yaw_cos
        [2]: speed_norm
        [3]: altitude_norm
        [4]: vspeed_norm
        [5]: target_dist_norm
        [6]: target_bearing_sin
        [7]: target_bearing_cos
        [8]: closure_rate_norm
        [9]: radar_visible_flag
        [10]: datalink_visible_flag
        [11]: reserved
    """
```

8.2 Critical Design: Raw Position Handling

The perception module uses RAW POSITIONS IN METERS for wingmen relative calculations:

- Clipped/normalized positions cause incorrect relative state computation
- CBVS controller needs accurate distances for phase detection
- Formation control requires precise inter-agent measurements

---

9. Reward Architecture

9.1 Hierarchical Reward System

Location: `training/environments/dcs_rewards.py`

```
r_total = w_form × r_form + w_pursuit × r_pursuit + w_safe × r_safe
```

9.2 Formation Rewards

Component	Description
Speed Coordination	Team speed variance penalty
Geometry Reward	Lateral spacing and cohesion
Velocity Alignment	Heading coordination
Wedge Formation	Slot error with persistence bonuses

9.3 Pursuit Rewards

Component	Description
Target Proximity	Zone-based standoff reward
Closure Rate	Approach/maintenance reward
Bearing Alignment	Nose-on-target reward

9.4 Standoff Zone Model

Zone	Distance	Reward
CRITICAL	<3km	Severe penalty
DANGER	3-4km	Moderate penalty
CLOSE	4-6km	Small penalty
OPTIMAL	**6-10km**	**Maximum reward**
FAR	10-25km	Reduced reward
TOO FAR	>25km	No reward

9.5 Phase-Based Weighting

Phase	Formation	Pursuit	Safety
INTERCEPT	0.35	0.65	0.4
APPROACH	0.50	0.50	0.5
STEADY	0.70	0.30	0.6
FORMATION	0.85	0.15	0.7

---

10. Safety Architecture

10.1 Safety Shield

Location: `training/safety/safety_shield.py`

DO-178C Level A Compliant Design with 4 layers:

Layer	Function
1. Input Validation	NaN/Inf detection, type coercion
2. Flight Envelope	Altitude/speed bounds, heading wrap
3. Rate Limiting	Heading/speed/altitude rate limits
4. Fail-Safe Recovery	Last known safe command fallback

10.2 Rate Limit Profiles

Mode	Heading	Speed	Altitude
DEFAULT	1.0×	1.0×	1.0×
PURSUIT	0.20×	0.50×	1.0×
JOIN	1.0×	2.0×	1.0×
DETOUR	0.16×	0.58×	0.30×
EMERGENCY	3.0×	2.5×	3.0×

10.3 Envelope Bounds

Parameter	Min	Max
Altitude	30m	12,000m
Speed	100 m/s	250 m/s
Heading Rate	3°/cmd	-
Speed Rate	4 m/s/cmd	-
Altitude Rate	15 m/cmd	-

---

11. DCS gRPC Interface

11.1 Client Architecture

Location: `dcs_interface/grpc/client.py`

```
┌─────────────────────────────────────────────────────────────────┐
│                          DCSGrpcClient                           │
├─────────────────────────────────────────────────────────────────┤
│ ┌──────────────────┐ ┌──────────────────┐ ┌───────────────────┐ │
│ │ Connection Mgmt  │ │  Service Stubs   │ │ Lua Code Generator│ │
│ │ ──────────────── │ │ ──────────────── │ │ ───────────────── │ │
│ │ • connect()      │ │ • metadata       │ │ • eval_lua()      │ │
│ │ • close()        │ │ • mission        │ │ • generate_hsa_lua()│
│ │ • reconnect()    │ │ • unit           │ │ • generate_state()│ │
│ │ • health_check   │ │ • controller     │ │ • sanitize_input()│ │
│ └──────────────────┘ │ • custom         │ └───────────────────┘ │
│                      │ • trigger        │                       │
│                      │ • coalition      │                       │
│                      └──────────────────┘                       │
└─────────────────────────────────────────────────────────────────┘
```

11.2 HSA Command Structure

```python
@dataclass
class HSACommand:
    unit_name: str          # DCS unit name
    heading_deg: float      # 0-360, 0=North, clockwise
    speed_mps: float        # Meters per second
    altitude_m: float       # Barometric altitude
    distance_m: float       # Waypoint distance (default 4000m)
```

11.3 Channel Configuration

Option	Value	Purpose
keepalive_time_ms	20000	Ping interval
keepalive_timeout_ms	10000	Ping ACK timeout
max_message_length	16 MB	Message size limit

---

12. HGAT-MAPPO-WM Architecture

12.1 4-Layer Stack Overview

The planned HGAT-MAPPO-WM architecture consists of four integrated layers:

```
┌─────────────────────────────────────────────────────────────────┐
│                     HGAT-MAPPO-WM 4-LAYER STACK                   │
├─────────────────────────────────────────────────────────────────┤
│                                                                  │
│ ┌─────────────────────────────────────────────────────────────┐  │
│ │ LAYER 4: COUNTERFACTUAL CRITIC                              │  │
│ │ • Centralized critic Q(s, a₁, a₂, a₃)                       │  │
│ │ • Counterfactual baseline: Q(s, a₋ᵢ, â)                     │  │
│ │ • Credit assignment: A^CF_i = Q(s,a) - Q(s, a₋ᵢ, π(o_i))    │  │
│ └─────────────────────────────────────────────────────────────┘  │
│                              ▲                                   │
│ ┌─────────────────────────────────────────────────────────────┐  │
│ │ LAYER 3: LEADER-FOLLOWER HIERARCHICAL POLICY                │  │
│ │ • Policy Selector: π_sel(mode | global_state)               │  │
│ │ • Leader Policy: π_L(a_L | o_L, mode, formation_state)      │  │
│ │ • Follower Policy: π_F(a_F | o_F, mode, leader_action)      │  │
│ └─────────────────────────────────────────────────────────────┘  │
│                              ▲                                   │
│ ┌─────────────────────────────────────────────────────────────┐  │
│ │ LAYER 2: HIERARCHICAL GRAPH ATTENTION NETWORK (HGAT)        │  │
│ │ • Level 1: Inter-Agent GAT (local neighborhood)             │  │
│ │ • Level 2: Inter-Group Transformer (global context)         │  │
│ │ • CommFormer: Learnable communication topology              │  │
│ └─────────────────────────────────────────────────────────────┘  │
│                              ▲                                   │
│ ┌─────────────────────────────────────────────────────────────┐  │
│ │ LAYER 1: WORLD MODEL (DREAMER V3)                           │  │
│ │ • RSSM: h_t = f(h_{t-1}, z_{t-1}, a_{t-1})                  │  │
│ │ • Encoder: z_t ~ q(z_t | h_t, o_t)                          │  │
│ │ • Formation Decoder: predict slot errors, quality           │  │
│ │ • Imagination Training: 15-step rollouts                    │  │
│ └─────────────────────────────────────────────────────────────┘  │
│                                                                  │
└─────────────────────────────────────────────────────────────────┘
```

12.2 Implementation Status

Component	Location
Layer 1: World Model	`training/marl/world_model/`
Layer 2: HGAT	`training/marl/hgat_mappo/`
Layer 3: Leader-Follower	`training/marl/hgat_mappo/`
Layer 4: Counterfactual	`training/marl/hgat_mappo/`
Hybrid Controller v2.0	`training/guidance/formation_cbvs_v2.py`
Config Infrastructure	`training/marl/hgat_mappo/config.py`

12.3 Training Curriculum (4 Phases)

Phase	Objective	Success Criteria
1	World Model Pre-training	Reconstruction MSE < 0.01
2	Formation-Only in Imagination	Formation quality > 0.95
3	Joint Formation and Pursuit	Formation > 0.8, Pursuit > 70%
4	Adversarial Robustness	Pursuit > 90%, Robust to 20 m/s wind

---

13. Data Flow and Pipeline

13.1 Step Execution Flow

```
┌─────────────────────────────────────────────────────────────────┐
│                        ENV.STEP() DATA FLOW                       │
├─────────────────────────────────────────────────────────────────┤
│                                                                  │
│  1. FETCH STATE FROM DCS                                         │
│     └─► get_state_batch(unit_names) → transforms, target_tr      │
│                                                                  │
│  2. UPDATE FSMs                                                  │
│     ├─► Aircraft FSM: update_state(telemetry) → aircraft_state   │
│     └─► Control FSM: update_mode(context) → control_mode         │
│                                                                  │
│  3. CHECK DETECTION                                              │
│     └─► _check_s300_detection() → detection_ok, target_uv        │
│                                                                  │
│  4. COMPUTE OBSERVATIONS                                         │
│     └─► dcs_perception.build_observations() → obs[N, 48]         │
│                                                                  │
│  5. POLICY / TEACHER ACTION                                      │
│     ├─► RL Policy: policy(obs) → actions[N, 3]                   │
│     └─► Teacher: cbvs_controller.compute_actions() → actions     │
│                                                                  │
│  6. COMMAND PIPELINE                                             │
│     ├─► handle_low_fuel_and_triad()                              │
│     ├─► prepare_control_and_uncertainty()                        │
│     ├─► build_planned_commands()                                 │
│     │   ├─► RTA Filter                                           │
│     │   ├─► Join/Recall Override                                 │
│     │   ├─► Cohesion Filter                                      │
│     │   ├─► WEZ Routing                                          │
│     │   └─► Safety Shield                                        │
│     └─► send_hsa_batch() → DCS gRPC                              │
│                                                                  │
│  7. COMPUTE REWARDS                                              │
│     ├─► Formation rewards                                        │
│     ├─► Pursuit rewards                                          │
│     └─► Safety rewards                                           │
│                                                                  │
│  8. RETURN (obs, rewards, dones, infos)                          │
│                                                                  │
└─────────────────────────────────────────────────────────────────┘
```

13.2 Command Pipeline Stages

Stage	Function	Purpose
1. Pre-processing	`handle_low_fuel_and_triad()`	Retirement & replacement
2. State Management	`update_reacquisition_state()`	REACQ lifecycle
3. Command Building	`build_planned_commands()`	Per-agent HSA
4. Transmission	`send_hsa_batch()`	gRPC dispatch

---

14. Configuration and Parameters

14.1 Key Formation Parameters

Parameter	Value	Description
wedge_trail_m	500.0	Trail offset behind leader
wedge_lateral_m	600.0	Lateral offset from leader
d_optimal_m	8000.0	Optimal standoff distance
d_min_m	5000.0	Minimum standoff
d_max_m	30000.0	Maximum tracking range

14.2 ATCFC Parameters

Parameter	Value	Description
turn_rate_threshold	0.005	Turn detection threshold (rad/step)
turn_rate_max	0.05	Maximum turn rate
blend_straight	0.1	Slot-tracking blend in straight flight
blend_turn_max	0.5	Slot-tracking blend at max turn
side_bias_gain	0.35	Side-keeping strength
trff_gain	0.8	Turn rate feed-forward gain

14.3 S-300 Detection Parameters

Parameter	Value	Description
s300_stability_steps	100	Consecutive steps for confirmation
ewr_detection_range_m	200000.0	Fallback detection range
ewr_use_range_fallback	False	Enable range-based fallback

---

15. Appendices

A. File Structure

```
cg-xmarl-3uav-dcs/
├── config/
│   └── __init__.py              # Configuration exports
├── dcs_interface/
│   └── grpc/
│       ├── client.py            # DCS gRPC client
│       └── generated/           # Protocol buffer stubs
├── training/
│   ├── environments/
│   │   ├── dcs_multi_agent_env.py  # Core environment
│   │   ├── dcs_perception.py       # Observation construction
│   │   ├── dcs_commands.py         # Command pipeline
│   │   ├── dcs_rewards.py          # Reward computation
│   │   └── dcs_threats.py          # Threat zone handling
│   ├── guidance/
│   │   ├── formation_cbvs_v2.py    # CBVS controller
│   │   └── join.py                 # WEZ-aware join heading
│   ├── fsm/
│   │   ├── aircraft_state_fsm.py   # Aircraft lifecycle FSM
│   │   └── control_mode_fsm.py     # Control mode FSM
│   ├── safety/
│   │   ├── safety_shield.py        # HSA command filtering
│   │   └── cohesion.py             # Cohesion filter
│   └── marl/                       # HGAT-MAPPO (planned)
│       ├── hgat_mappo/
│       │   ├── config.py           # HGATMAPPOConfig
│       │   └── __init__.py
│       ├── world_model/
│       └── curriculum/
├── scripts/
│   ├── collect_formation_data.py   # Data collection
│   └── restart_mission_webgui.py   # WebGUI automation
└── guides/
    ├── HGAT-MAPPO-WM-MASTER-INDEX.md
    ├── HGAT-MAPPO-WM-ARCHITECTURE-PART1.md
    ├── HGAT-MAPPO-WM-ARCHITECTURE-PART2.md
    └── HGAT-MAPPO-WM-IMPLEMENTATION-PLAN.md
```

B. Critical Bug Fixes Applied

Date	Issue	Fix
2025-12-20	Wedge asymmetry during turns	Consistent heading reference frames
2025-12-24	Zigzag oscillation	Reduced lateral correction gains
2025-12-24	Inside wing double deviation	Smooth side_bias ramping
2026-01-02	Replacement triad S-300 gate	Keep s300_detection_confirmed=True

C. Environment Variables

Variable	Default	Description
DCS_DEBUG_CBVS	0	Enable CBVS debug logging
DCS_DEBUG_FSM	0	Enable FSM debug logging
CGX_LOW_FUEL_FRAC	0.15	Low fuel threshold override

Document End — CG-XMARL v4.1 Technical Specification