Forest Navigation UAV

Learning-Based Navigation in Dense Forests

Thodoris Evangelakos
Autonomous Agents - INF412
27-02-2026

The Problem

Autonomous UAV navigation in dense forests.

Goal: - Reach a target location
- As fast as possible
- Without collisions

Particular challenges: - Narrow passages and occlusions
- Partial observability (LiDAR only)
- High-speed control under real-time constraints
- Collision is catastrophic

Success Criteria

We define success strictly:

- **Success:** Reach goal within 30s without collision
- **Collision:** Any contact with obstacle
- **Time-to-goal:** Measured only for successful runs
- **Minimum clearance:** Smallest distance to obstacle
- **Shield intervention rate:** % steps safety layer modified action

Speed is desired but safety is non negotiable.

System Architecture

Closed-loop pipeline:

- Sensors

- Observation vector
- SAC Policy
- Safety Shield
- Command
- Simulator / Gazebo

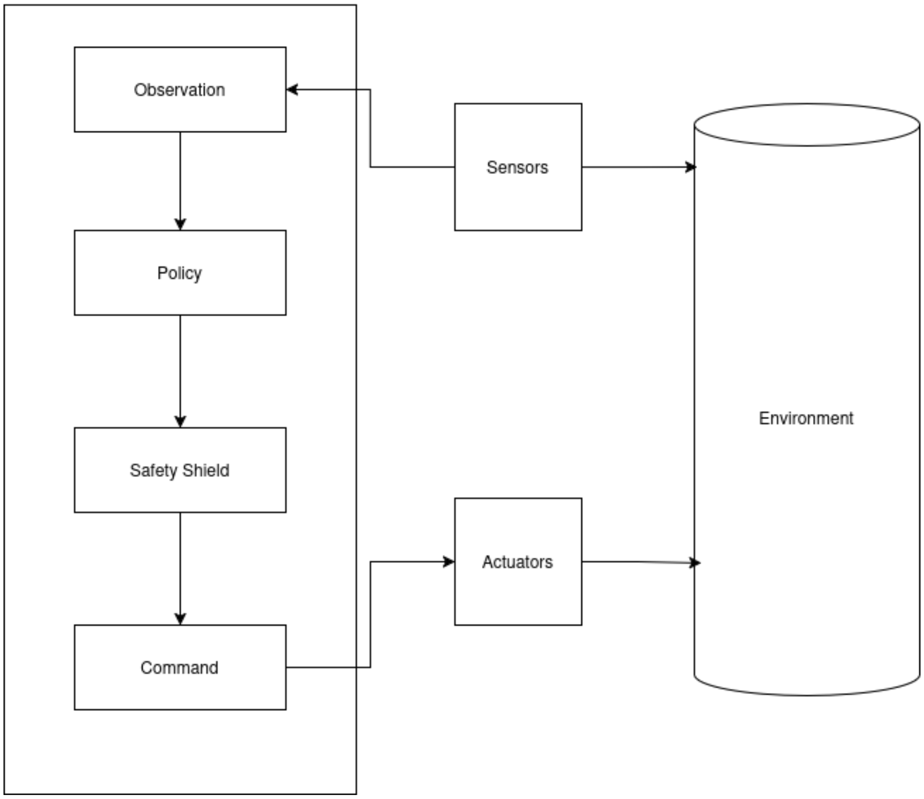


Figure 1: Pipeline

Key idea: Learning handles complexity.
Shield enforces safety.

Why This Approach?

Why Learning?

- Forests have structure and patterns
- Hard to hand-engineer every case
- Fast runtime inference

Why SAC?

- Continuous control
- Stable training
- Good sample efficiency

Why a Safety Shield?

- Learned policies are not guaranteed safe
- Shield = deterministic last line of defense (usually)

Training Strategy

Train in custom fast in-memory simulator (`fastsim`)

Why? - Orders of magnitude faster than Gazebo
- 16 parallel environments
- 10M timesteps easily achievable, a lot is more feasible without time constraints of the deadline

Then validate in: - Gazebo + ROS2
- Procedurally generated forest worlds
- Higher density than real-life forests (2x-10x as dense according to literature)

Observations and Actions

Observation (96D vector)

- 90 LiDAR beams (normalized)

- cos(goal angle), sin(goal angle)
- normalized distance
- normalized speed, yaw rate
- normalized height error

Action

- Forward acceleration
- Yaw acceleration
- Vertical acceleration
  Scaled to physical limits
  Filtered by safety shield

Reward Design (High-Level)

Reward encourages:

+ Progress toward goal
+ Speed aligned with goal
- Proximity to obstacles
- Per-step penalty
- Large collision penalty
- Stalling and/or truncating

Key lesson: Reward shaping matters, otherwise the UAV plays mary-go-around/kamikazes onto trees for no apparent reason (empty field everywhere else)

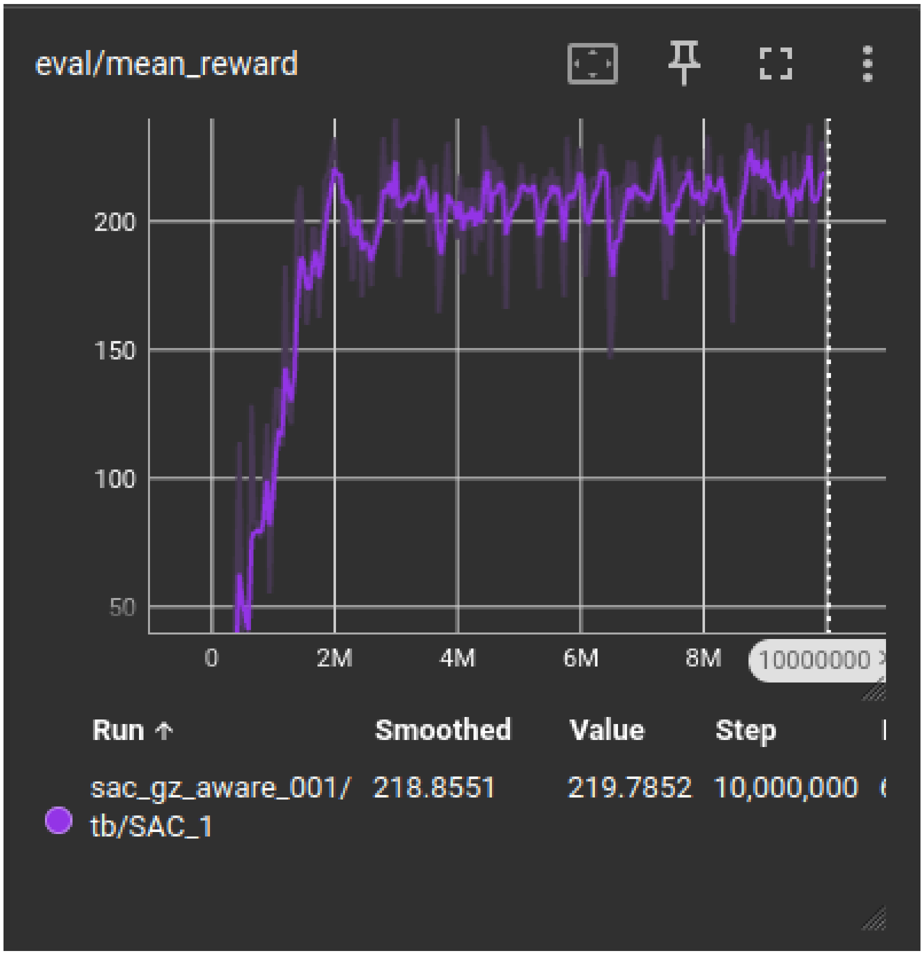
Results

Final Performance

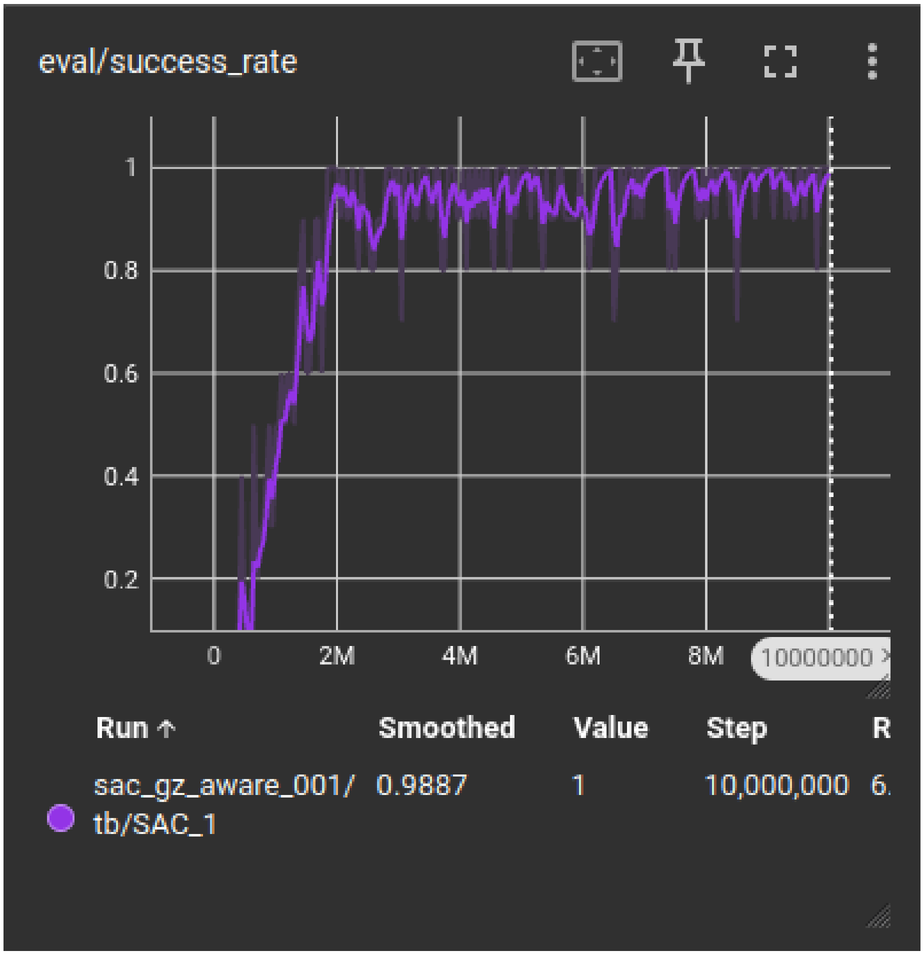
- **85% success rate**
- **11.95% collision rate**
- **Median time to goal: 12.35s**
- **Minimum clearance: 2.806m**
- **Shield interventions: 8.43%**

Learning Curves

Reward curve
Success rate
Collision rate
Shield activation



Run ↑	Smoothed	Value	Step
● sac_gz_aware_001/tb/SAC_1	218.8551	219.7852	10,000,000



	Run ↑	Smoothed	Value	Step	R
●	sac_gz_aware_001/ tb/SAC_1	0.9887	1	10,000,000	6.

(Show only 1–2 most important plots during presentation.)

Demo

Demo video: Relevant button in frontpage

Key observations: - Controlled navigation
- Shield prevents imminent collisions
- Smooth goal convergence
- Very low likelyhood of getting stuck between trees

Engineering Challenges

1) Reward Hacking

Agent farmed speed rewards by circling/going around the forest.

Fix: - Virtual barrier around the map
- Massive increase in success reward

Result: No more circling.

---

2) Sim-to-Sim Transfer Gap

Policy trained in fastsim failed in Gazebo.

Cause: - Oversimplified dynamics
- Sensor mismatch

Fix: - Matched configurations
- Froze height, roll, pitch initially
- Refined action interface

---

3) Computational Bottleneck

Training too slow.

Cause: - Raycasting and collision checks CPU-bound

Fix: - Hash grid for trees
- Lazy collision checking
- Parallelized stepping

Result: Reasonable training times.

---

4) Unrealistic Dynamics

Velocity commands caused “Flying saucer behavior”.

Fix: - Switched to acceleration control
- Added inertia realism

Transfer improved.

What Worked

- SAC handled continuous control well
- Shield enabled aggressive but safe motion
- Domain randomization improved robustness
- Fast simulator enabled rapid iteration

Limitations

- Sim-to-sim gap still present
- Shield can be overly conservative
- Dynamics still simplified
- No wind, no complex aerodynamics
- No multi-agent behavior yet

Future Work

- Improve physics realism
- Refine safety shield (less conservative)
- Train in denser, more complex forests
- Add vision-based perception
- Multi-agent coordination
- Real UAV testing

Takeaways

- Learning + hard safety layer is powerful

- Fast training infrastructure is essential
- Reward design and transfer are critical
- Promising results, but not yet real-world ready

Thank You

Questions?