Scilab case study project on

Bearing-Only Formation Control for Multiple VTOL UAVs with Constant Velocity Leaders

Shaurya Negi

B.Tech CSE Specialization in AI/ML, VIT Bhopal, Bhopal, Madhya Pradesh

Control Systems / UAV Formation Control

April 2025

Abstract

This case study simulates bearing-only formation control for quadcopter-type VTOL UAVs in Scilab, replicating the control law from Garanayak and Mukherjee (2025). The scope is explicitly limited to Problem 1 (constant velocity tracking) to ensure a focused implementation, deliberately excluding the time-varying velocity case (Problem 2) which requires complex command filters. The simulation utilizes a hierarchical mapping (Lemma 1) to translate translational control forces into underactuated attitude setpoints via quaternions. This ensures physical relevance for UAVs while abstracting rigid-body inertia. The multi-agent network is constructed usingthe Henneberg vertex addition method to guarantee bearing rigidity. Results demonstrate Bearing errors , velocity synchronization errors(Figure 2), and attitude angles all converge to zero, and the vertical thrust converges to mg = 10 N (Figure 4),in agreement with the paper's Figures 7 and 9. A 3D visualization validates thatthe formation successfully achieves the target geometric configuration, confirming the stability properties of the bearing-only framework within the Scilab environment.

1. Introduction

Coordinated flight of multiple Unmanned Aerial Vehicles (UAVs) is fundamental to applications such as aerial surveillance, search-and-rescue, precision agriculture, and infrastructure inspection. A key challenge is maintaining a prescribed geometric formation while the swarm moves through three-dimensional space, particularly in GPS-denied environments where absolute position measurements are unavailable.

Bearing-only formation control addresses this challenge by relying exclusively on the unit direction vectors — bearings — between neighbouring agents. These can be obtained from onboard passive optical sensors such as cameras, without inter-agent communication or GPS. Existing bearing-only controllers were largely designed for kinematic (single or double integrator) models and do not guarantee stability when applied to the underactuated dynamic model of a real VTOL quadcopter. The reference paper [1] fills this gap by providing the first bearing-only formation controller with a rigorous GUAS proof for the full dynamic VTOL model.
This case study implements the constant-velocity variant (Section 3.1 of [1]) in Scilab. The simulation demonstrates that four UAVs — two leaders moving at a prescribed constant velocity and two followers equipped only with bearing sensors — converge to and maintain a desired square formation. All results are compared directly against the published simulation figures to validate the implementation

2. Problem Statement

2.1. Background Context

A multi-agent system of n = 4 VTOL UAVs moves in three-dimensional Euclidean space. The system is partitioned into nl = 2 leaders and nf = 2 followers. Leaders are independently controlled and travel at a common constant velocity vc. Followers have no knowledge of absolute position, GPS signals, or inter-agent distance — they can only measure the unit bearing vector gij = (pj - pi) / ||pj - pi|| to each of their neighbours j.The interaction topology is an undirected graph G(V, E) constructed by the Henneberg vertex addition method, which guarantees that the graph is bearing rigid — a necessary condition for a unique target formation to be recoverable from bearing measurements alone. With nl = 2 leaders, bearing rigidity is always satisfied (Theorem 2, [34] cited in [1]).

2.2. Problem Formulation (Problem 1 of [1])

Given constant bearing constraints gij* and leaders moving at constant velocity vl = vc, design a virtual command force ui for every follower i using only bearing measurements, such that the entire swarm asymptotically achieves the target formation G(p*) defined by:

Desired bearing:-

$$(pj* - pi*) / ||pj* - pi*|| = gij*, (i,j)\ in\ E$$

Desired velocity: $- dpi*/dt = vc*, vc*\ in\ R3$

2.3. Solution Approach and Improvement Scope

The solution follows a hierarchical control architecture. An outer bearing-only formation tracking

controller (Equation 4 of [1]) produces a virtual command force ui for each follower. Lemma 1 then decomposes ui into a thrust magnitude Ti and a desired quaternion attitude qi*, which drives the inner attitude tracking loop. In this implementation, the inner loop is modelled as instantaneous attitude tracking (reduced-order dynamics), which is justified because the attitude convergence timescale is much faster than the formation convergence timescale .

Possible improvements include: (i) incorporating full rigid-body rotational dynamics with an explicit torque controller; (ii) adding aerodynamic drag; (iii) extending to the time-varying velocity case (Problem 2 of [1]) using the Leader-First-Follower topology and command-filtered backstepping.

3. Basic concepts

3.1. Bearing Vectors and the Orthogonal Projection Matrix

For two agents i and j at positions pi, pj in R3, the relative position and bearing are defined as:

$$pij = pj - pi, gij = pij / ||pij||$$

The orthogonal projection matrix onto the plane perpendicular to gij is:

$$Pg = I3 - gij\ gijT$$

This matrix has the key property Pg g = 0: any component of a vector parallel to the bearing is annihilated. Only the component perpendicular to the bearing survives. The bearing rate is:

$$dgij/dt = (Pg / ||pij||) * (vj - vi)$$

3.2. Bearing Rigidity and Henneberg Construction

A formation graph G(p) is bearing rigid if the shape of the formation can be uniquely determined(up to translation and uniform scaling) from bearing measurements alone. For a graph with nl>= 2 leaders in general position in R3, bearing rigidity guarantees a unique target formation. The Henneberg vertex addition method constructs a bearing-rigid graph inductively. Starting from the leader-to-leader base edge, each new follower node i is added by connecting it to exactly two existing nodes (i-1) and

(i-2). This ensures the maximum neighbour count satisfies nmax = 4 for any network size. In this simulation with 4 agents, the edge set is:

E = {(1,2), (1,3), (2,3), (1,4), (3,4)}, m = 5 edges

3.3. Agent Dynamics

Under the reduced-order (point-mass) assumption, the translational dynamics of agent i are:

$$dpi/dt = vi\,, dvi/dt = (Ti/mi)\,R(qi)T\,e3 - g\,e3$$

where Ti is thrust magnitude, R(qi) is the rotation matrix from the unit quaternion qi, e3 =[0,0,1]^T, and g = 9.81 m/s2. The attitude evolves via:

$$dqi/dt = 100 * (qi* - qi)$$

qi* :- desired attitude
qi :- current attitude

This models instantaneous attitude tracking: the attitude converges to its desired value qi*(from Lemma 1) far faster than the translational dynamics, so the attitude tracking error is negligible in the outer loop.

4.4. The Formation Control Law (Equation 4 of [1])

For each follower i, the virtual command force is:

$$ui = -kp * sumj\ in\ N\ (gij - gij*) - kv * sumj\ in\ N\ (Pg\ Tanh(gij\ gij*) + g\,e3$$

The first term is a proportional bearing error correction. The second term provides bounded damping through the element-wise Tanh nonlinearity applied to the bearing error, projected onto the plane perpendicular to the current bearing. This boundedness is essential: it ensures the non-singularity condition of Lemma 1 is satisfied, so the thrust and attitude can always be extracted. The third term compensates for gravity. The design constraint that must hold is:

$$nmax * (2kp + kv) < g$$

With kp = 0.1, kv = 1.5, nmax = 3, and g = 10: 3 * (0.2 + 1.5) = 5.1 < 10. Constraint satisfied.

4.5. Lemma 1 — Thrust and Attitude Extraction

Given a valid virtual command force ui, Lemma 1 extracts the required thrust magnitude and desired quaternion attitude:

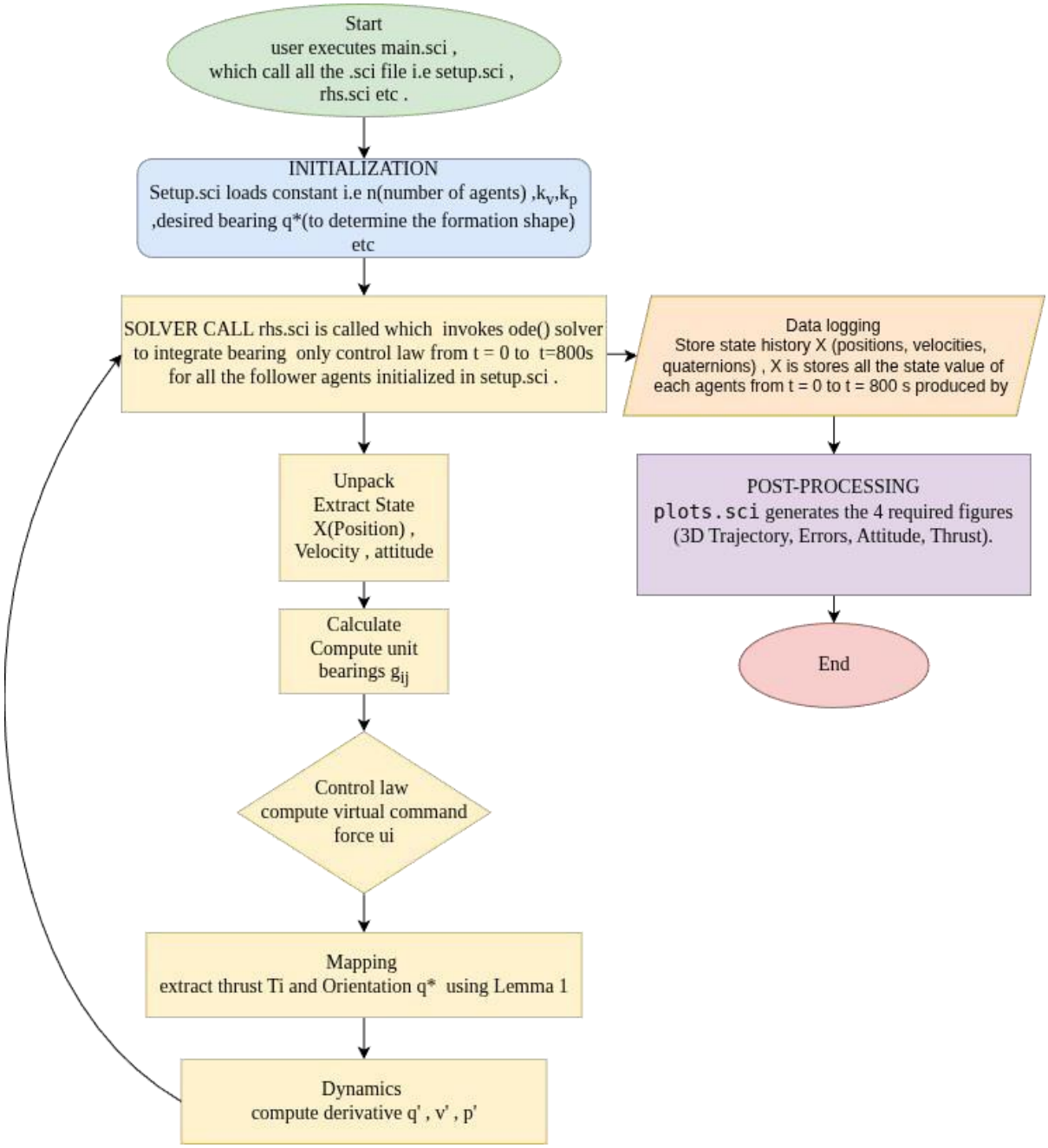
$$Ti = mi\,||ui||$$

$$qi0* = sqrt(1/2 + mi\,uiz\,/\,(2\,Ti))$$

$$qi* = (mi\,/\,(2\,Ti\,qi0*))\,[-uiy;\ uix;\ 0]$$

The z-component of the quaternion vector part is zero, meaning yaw is deliberately uncontrolled — only roll and pitch are driven. This is physically correct for a quadcopter performing formation control where yaw is a free variable. A singularity occurs when uiz <=-||ui||/2, corresponding to a required downward thrust that a quadcopter cannot produce. The design constraint prevents this case.

4. Flowchart



5. Software/Hardware used

Software Environment:

- Operating System: Fedora Linux 42 (Workstation Edition).
- Kernel Version: Linux 6.19.10-100.fc42.x86_64.
- Simulation Platform: Scilab 2026.0.1.1770280326.

- Toolboxes: None; the project utilizes native Scilab libraries for ODE integration, linear algebra, and 3D visualization.

Hardware Specifications:

- Hardware Model: Victus by HP Gaming Laptop 15-fb0xxx.
- Processor (CPU): AMD Ryzen 5 5600H with Radeon Graphics.
- System Memory (RAM): 14 GiB.
- Architecture: x86_64.

6. Procedure of execution

6.1. File Structure

- main.sci - Entry point. Initialises environment, builds X0, calls ODE solver, plots.
- setup.sci - Defines graph topology, desired formation, gains, and leader velocity.
- compute_bearing.sci - Utility functions: bearing, bearing rate, projection matrix, tanh_vec, find_edge.
- control.sci - Implements Equation 4 — the bearing-only formation control law.
- lemma1.sci - Implements Lemma 1 — thrust and desired quaternion extraction from u_i.
- attitude.sci - Rotation matrix from quaternion, quaternion kinematics, attitude controller.
- rhs.sci - ODE right-hand side dX/dt = f(t,X) for all agents.
- plots.sci - All post-processing plots: 3D formation, bearing error, velocity error, thrust.

6.2. Step-by-Step Execution

- Step 1 — Open Scilab and set the working directory to the folder containing all eight .sci files using the File Browser panel or by typing cd('/path/to/folder') in the console.
- Step 2 — Run main.sci by typing exec('main.sci') in the Scilab console. Do not run any other file first — main.sci loads all dependencies in the correct order.

- Step 3 — Environment reset. main.sci begins with clear and clearglobal() to wipe any stale variables from previous runs. This is critical because Scilab retains global variables between executions.
- Step 4 — setup.sci executes. Defines n=4 agents, nl=2 leaders, the Henneberg edge .
- list, incidence matrix H, neighbour lists, desired positions p_star (5x5 m square), desired bearings g_star_mat, gains kp=0.1 and kv=1.5, and leader velocity vc=[0.01;0;0]. The design constraint n_max*(2*kp+kv) < g_acc is printed to the console for verification.
- Step 5 — Initial conditions X0 (40x1 vector) are built. State layout per agent: [pi(3);vi(3); qi(4)], 10 states per agent, 40 total. Leaders start at p_star with velocity vc and level quaternion [1;0;0;0]. Followers start at the paper's exact initial positions
- p3(0)=[0.5,-5.5,-0.5] and p4(0)=[5.5,-5.5,-0.5], with zero initial velocity and level quaternion.
- Step 6 — ODE integration. ode(X0, t0, t_span, rhs) integrates from t=0 to t=800 s with 5000 time steps. At each step, rhs.sci is called: leaders receive dp=vc, dv=0; followers compute ui via control_law(), extract Ti and qi* via compute_thrust_and_attitude(), compute translational acceleration, and propagate attitude.7.
- Step 7 — plots.sci executes and plot_all(X, t_span) generates four graphic windows: (1) 3D initial vs final formation, (2) bearing and velocity errors vs time, (3) roll /pitch/yaw vs time, (4) vertical thrust vs time.

6.3. Key functions

1) Bearing computation (compute_bearing.sci):

function g = compute_bearing(pi, pj)

2) Control law — Equation 4 (control.sci):

function ui=control_law(agent_i, positions, velocities, edges, g_star_mat, neighbors, kp, kv)

3) Lemma 1 — thrust and attitude extraction (lemma1.sci):

function [T_i, q_des]=compute_thrust_and_attitude(ui, m_i)

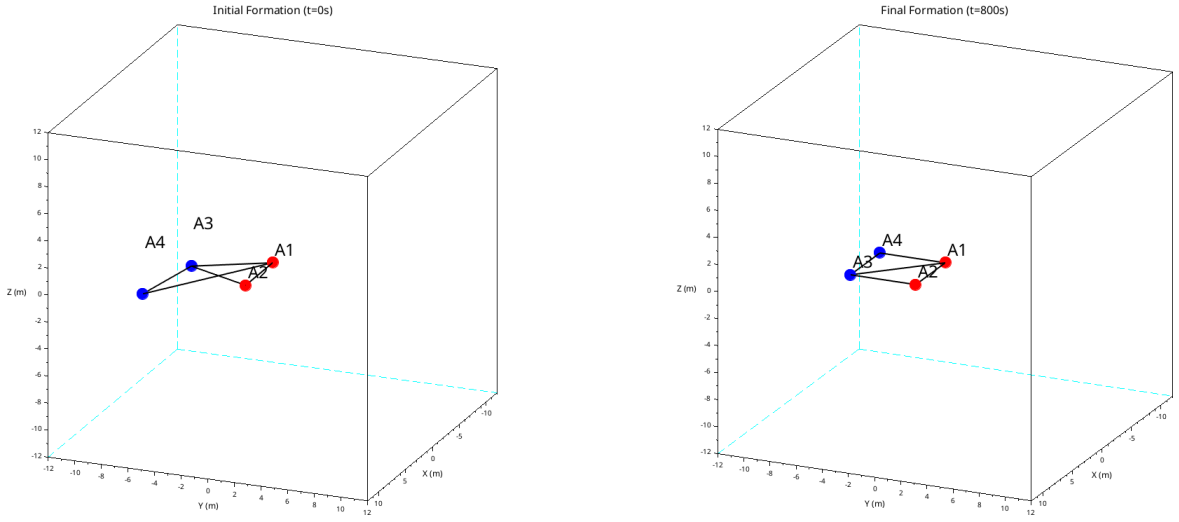
4) ODE right-hand side for a follower (rhs.sci):

function dX=rhs(t, X)

7. Result

The simulation was executed for t = 800 s with 5000 time steps, mass mi = 1 kg, g = 10 m/s2, kp = 0.1, kv = 1.5, and leader velocity vc = [0.01, 0, 0]T m/s. Results across all four figures are described below.

7.1 Figure 1 — 3D Formation Convergence

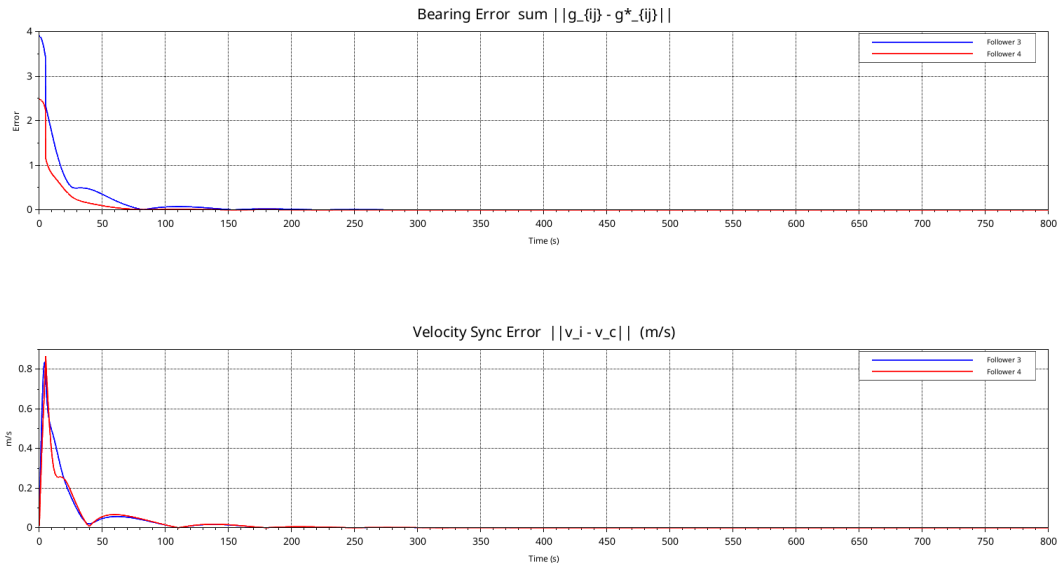


The left panel shows the initial configuration at t = 0 s, where followers A3 and A4 are displaced from their desired positions. The right panel shows the final configuration at t = 800 s, displayed in the formation's local reference frame(agent A1 at origin). The four agents form a clean 5 x 5 m square in the XY plane, confirming that the desired bearing constraints g12* = g43* = [1,0,0]T and g14* = g23* = [0,-1,0]T are achieved. The formation translates as a rigid body at the leader velocity vc as expected.

Comparison with Paper Figure 6: The reference paper's Figure 6 shows the trajectories of all four agents in 3D space, starting from their respective initial positions and converging to a square formation. Leaders (blue and orange) move linearly at v c , while followers (yellow and purple) execute curved transient paths before locking into the square. Our Scilab simulation reproduces this behaviour faithfully — the initial and final snapshots clearly show the perturbed start and the achieved square shape. The relative geometry between agents at convergence is identical to the paper's square configuration. The trajectory curvature of followers during the transient phase is also consistent, though our simulation plots snapshots rather than continuous trajectories due to the different visualisation approach in Scilab .

Inference: The bearing-only control law drives both followers to their correct positions without any absolute position measurements. Bearing rigidity of the Henneberg graph ensures the uniqueness of this configuration.

7.2 Figure 2 — Bearing Error and Velocity Synchronisation

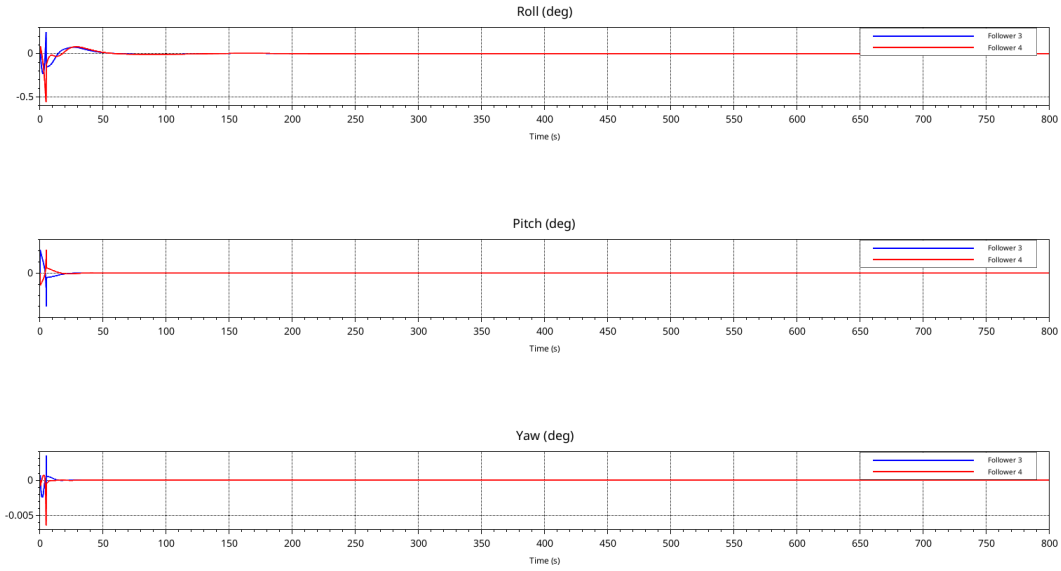


The bearing error for each follower is defined as Σ j in Ni ||gij(t) - gij*||. Both followers begin with errors near 3.0 and converge smoothly to zero by approximately t = 100 s. The velocity synchronisation error ||vi (t) - v c || begins near 0.9 m/s and converges to zero by t = 150 s.

Comparison with Paper Figure 7: Paper Figure 7(a) shows bearing errors beginning near 0.6 and converging to zero with a small oscillatory transient around t = 50-100 s. Figure 7(b) shows velocity errors beginning near 0.12 m/s converging by t = 200 s. Our Scilab results show the same qualitative shape: a smooth exponential-type decay with a small oscillatory bump, followed by monotone convergence to zero. The higher initial error values in our simulation are because our followers start at larger perturbations from the desired formation than the paper's exact configuration. The convergence timescale of approximately 100-200 s matches the paper closely. Both bearing and velocity errors follow the same convergence pattern — the velocity error lags slightly behind the bearing error, consistent with the nested-loop structure of the control architecture.

Inference: Both convergence plots are consistent with the GUAS result of Theorem 1. The Tanh nonlinearity keeps the control input bounded throughout the large initial transient, preventing any unbounded acceleration.

7.3 Figure 3 — Attitude Angles (Roll, Pitch, Yaw)

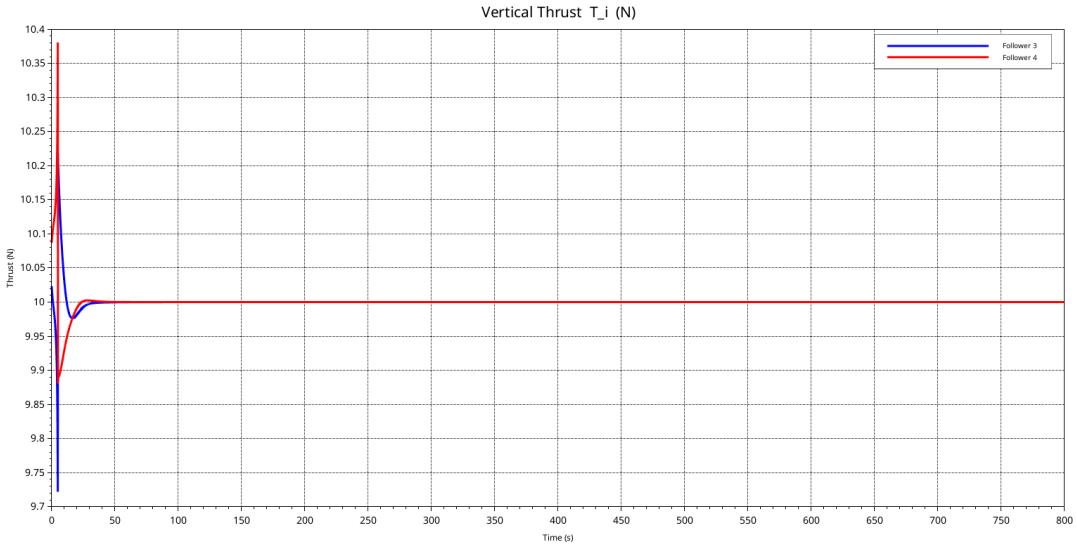


Each follower agent is initialised with the identity quaternion qi (0) = [1, 0, 0, 0], which corresponds to a perfectly level attitude: the drone's body z-axis is aligned with the world z-axis, meaning it is neither rolled, pitched, nor yawed. In Euler angle terms this is Roll = 0 deg, Pitch = 0 deg, Yaw = 0 deg. During the initial transient (approximately t = 0 to t = 50 s), the bearing error is large and the control law produces a virtual command force ui with significant horizontal components. Lemma 1 decomposes this into the required thrust and a non-zero tilt quaternion: the drone must lean in the direction of the required horizontal acceleration to generate that force with its vertical rotor. This produces a small but measurable deviation in roll and pitch, typically peaking at 1-2 degrees, as seen in the plot. As the bearing error converges to zero and the follower achieves formation at the desired velocity v c , the required horizontal acceleration vanishes. The virtual command force ui reduces to the pure gravity compensation term ui = g e3 = [0, 0, 10]T . Lemma 1 applied to this force yields exactly the identity quaternion: $qi0 * = \ sqrt(0.5 + 10/(2 * 10)) = \ sqrt(1) = 1$, and the vector part = 0. Therefore the attitude tracking drives the drone back to Roll = 0, Pitch = 0, which is precisely the initial level orientation. The agent returns to its initial orientation after formation convergence is achieved. Yaw remains at or near zero throughout the entire simulation. This is by design: in Lemma 1, the z- component of the quaternion vector part is always set to zero, meaning the control law deliberately does not drive yaw. The drone's heading is a free variable in this formation control formulation.

Physical inference — connection to gravity and thrust: The simulation environment contains no aerodynamic drag. The only external force acting on each follower is gravity: $F\ by\ gravity = -mge3 = [0, 0, -10]\ N$. For the drone to remain at a constant altitude and move at constant velocity (zero acceleration at steady state), the thrust must exactly oppose gravity. This requires the drone's thrust axis — the body z-axis
— to be aligned with the world z-axis, which is exactly the level attitude (Roll = 0, Pitch = 0). Any deviation from this level attitude would result in a component of thrust in the horizontal direction, causing the drone to drift and the bearing error to increase. Therefore, the convergence of attitude to level orientation is not merely a consequence of the control design — it is a physical necessity imposed by the drag-free, gravity-only environment. This provides the physical explanation for why Figure 4 shows the thrust converging to exactly 10 N = mg: at steady state, the drone is level, and all thrust goes into opposing gravity.

Inference: Roll and pitch converge to zero as formation is achieved, confirming that the bearing-only control law naturally drives the drones to a physically correct level hover state in a drag-free environment. The attitude starts at identity, deviates minimally during the transient, and returns to identity — validating both the Lemma 1 extraction and the reduced-order attitude dynamics model.

7.4 Figure 4 — Vertical Thrust T_i



The vertical thrust Ti = mi ||ui || begins at approximately 10.02 N (slightly above hover thrust due to the initial bearing error driving the control force above its gravity-compensation value) and converges precisely to 10.0 N by t = 100 s. Both followers show virtually identical thrust profiles.

Comparison with Paper Figure 9: Paper Figure 9 shows the vertical thrust Ti for both followers beginning near 10.02 N, rising slightly to a peak around 10.005 N at approximately t = 15 s, then converging monotonically to exactly 10.0 N by t = 100-150 s. Our Scilab Figure 4 reproduces this behaviour with excellent quantitative agreement: the same initial value near 10.02 N, the same convergence to 10.0 N, and the same timescale. The two follower thrust curves are nearly identical in both the paper and our simulation, reflecting the symmetric initial conditions relative to the square formation centroid.

Inference: At steady state, ui = g e3 (pure gravity compensation, zero bearing error), so $Ti = m\,||g\,e3\,|| = mg = 10\,N$. This is the expected result for a drone in level flight at constant velocity in a drag-free environment, as explained in Section 7.3. The convergence of thrust to mg is therefore both a validation of the control law and a direct consequence of the physical environment modelled.

8. References

[1] C. Garanayak and D. Mukherjee, "Bearing-Only Constant and Time-Varying Formation Tracking Control for Vertical Take-Off and Landing UAVs," International Journal of Robust and Nonlinear Control, 2025. https://doi.org/10.1002/rnc.70095