Subtask 2: Transition from Handwritten Name to “Happy New Year!” Formation

Student Name

1 Introduction and Problem Definition

Subtask 2 extends the static formation problem of Subtask 1 by requiring the swarm to *reconfigure* from an already-formed handwritten name to a new static holiday greeting (“Happy New Year!”). The swarm starts in the final configuration of Subtask 1 (positions close to the handwritten name) and must converge to the greeting formation while maintaining collision-free, smooth, physically feasible motion.

From the perspective of numerical modeling, this is again naturally formulated as an *Initial Value Problem (IVP)* for a coupled system of ODEs, where the initial condition is the terminal state of Subtask 1. Compared to Subtask 1, the key additional challenge is *assignment continuity*: drones should be mapped to new greeting target points so as to minimize travel distance and avoid large path crossings.

Inputs

- Drone swarm state at the end of Subtask 1: $\{x_i(0), v_i(0)\}_{i=1}^N$
- Greeting specification: image (or text fallback) representing “Happy New Year!”
- Number of drones N (fixed, consistent with Subtask 1)
- User-defined parameters: gains k_p, k_d, k_{rep}, safety radius R_{safe}, and speed limit $v_{\max}$

Outputs

- Trajectories $x_i(t)$ for all drones during the transition and hold period
- Visualization of the swarm converging from the name to the greeting

2 Mathematical Model

2.1 State Variables

Each drone $i \in \{1, \ldots, N\}$ is described by position $x_i(t) \in \mathbb{R}^2$ and velocity $v_i(t) \in \mathbb{R}^2$. The full swarm state is

$$Y(t) = (x_1, v_1, \ldots, x_N, v_N) \in \mathbb{R}^{4N}.$$

2.2 IVP Dynamics with Velocity Saturation

The governing equations are identical to Subtask 1, but with a *new static target set* $\{T_i\}_{i=1}^N$ representing the greeting:

$$\dot{x}_i = v_i \min\left(1, \frac{v_{\max}}{\|v_i\|}\right), \tag{1}$$

$$\dot{v}_i = \frac{1}{m}\Big[k_p(T_i - x_i) - k_d v_i + \sum_{j\neq i} f_{\text{rep}}(x_i, x_j)\Big], \tag{2}$$

with initial conditions provided by the final state of Subtask 1:

$$x_i(0) = x_{i,\text{name}}, \qquad v_i(0) = v_{i,\text{name}}.$$

Equation (**??**) enforces the speed limit $\|\dot{x}_i\| \le v_{\max}$, which is crucial for stable visualization and physical realism.

2.3 Collision Avoidance Force

Collision avoidance is modeled by an inverse-distance repulsive force activated inside a safety radius R_{safe}:

$$f_{\text{rep}}(x_i, x_j) = \begin{cases} k_{\text{rep}} \dfrac{x_i - x_j}{\|x_i - x_j\|^3}, & \|x_i - x_j\| < R_{\text{safe}}, \\ 0, & \text{otherwise.} \end{cases}$$

This formulation preserves symmetry (Newton's third law at the modeling level) and produces strong short-range separation without influencing far-away drones.

3 Greeting Target Extraction

3.1 Image Processing Pipeline

The greeting is provided either as (i) a raster image, or (ii) a text fallback rendered into an image. In both cases, a binary/edge representation is produced and N target points are sampled.

A typical pipeline is:

$$\text{Input image} \to \text{Grayscale} \to \text{Edge detection (Canny)} \to \text{Uniform point sampling.}$$

Let $\mathcal{E}$ be the set of extracted edge pixels. The greeting target set is

$$\{T_i^{\text{greet}}\}_{i=1}^N \subset \mathcal{E},$$

scaled and centered into the simulation canvas.

4 Assignment and Transition Strategy

4.1 Assignment Objective

The swarm must assign each drone to a unique greeting target point. The assignment cost matrix is

$$D_{ij} = \|x_i(0) - T_j\|,$$

and the ideal objective is to minimize total travel:

$$\min_{\pi \in S_N} \sum_{i=1}^N \|x_i(0) - T_{\pi(i)}\|.$$

4.2 Practical Assignment Method

To keep computation efficient for $N \approx 2000$, a greedy nearest-neighbor assignment is used. In practice, this yields smooth transitions because $x_i(0)$ is already structured (the handwritten name), and the greeting is similarly structured. Greedy assignment also supports *continuity*: the target assignment is recomputed from the current formation at the start of Subtask 2, reducing path crossings relative to a fixed assignment from the original grid.

4.3 Transition Timing

The simulation is run for a fixed number of transition frames followed by a hold phase. Let T_{sim} be the total transition time and Δt the integrator step. Then the number of RK4 steps is

$$N_{\text{steps}} = \frac{T_{\text{sim}}}{\Delta t}.$$

Multiple RK4 steps per rendered video frame are used to ensure smooth animation.

5 Numerical Method

5.1 RK4 Time Integration

The coupled ODE system (**??**)–(**??**) is integrated using the classical fourth-order Runge–Kutta scheme:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

where $y = (x, v)$ and

$$k_1 = f(t_n, y_n), \quad k_2 = f\big(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_1\big), \quad k_3 = f\big(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_2\big), \quad k_4 = f(t_n + h, y_n + hk_3).$$

RK4 is selected because the model contains strong restoring forces (k_p) and potentially stiff collision forces (repulsion), where lower-order methods (Euler/RK2) may require impractically small steps to avoid instability.

6 Verification and Validation

6.1 Formation Error Metric

A standard quantitative measure of convergence is the mean target error:

$$e(t) = \frac{1}{N}\sum_{i=1}^{N} \|x_i(t) - T_i\|.$$

Successful completion of Subtask 2 is indicated by $e(t)$ decreasing to a small steady value during the hold interval.

6.2 Safety Metric

Collision avoidance can be verified by tracking the minimum pairwise distance:

$$d_{\min}(t) = \min_{i \neq j} \|x_i(t) - x_j(t)\|.$$

A valid simulation should satisfy $d_{\min}(t) \geq R_{\text{safe}}$ for nearly all time (in practice, small transient violations may indicate parameter tuning is required).

7 Selected Code Segments and Their Meaning

This section connects representative implementation details to the model equations.

7.1 Greeting Target Extraction

The greeting can be loaded from an image file or rendered from text, then edge pixels are sampled into N points:

```
# targets = extract_points(PHASE2_IMAGE, NUM_DRONES)
# or a text fallback if the image is missing
# targets = extract_points(PHASE2_TEXT_FALLBACK, NUM_DRONES, is_text=
   True)
```

This corresponds to constructing $\{T_i\}_{i=1}^{N}$ from the edge set $\mathcal{E}$.

7.2 Assignment from Current Formation

At the start of Subtask 2, targets are assigned based on the current positions (end of Subtask 1):

```
assigned = greedy_assignment(current_positions, targets)
```

This implements the distance-based assignment objective with a greedy approximation.

7.3 Force Model and ODE Right-Hand Side

The acceleration combines attraction, damping, and repulsion:

```
spring = K_P * (tgt - pos)
damping = K_D * vel
rep = repulsion(pos)
dv = (spring + rep - damping) / M
```

which corresponds directly to Eq. (**??**).

7.4 RK4 Integration Loop

The swarm is advanced by repeated RK4 steps, with multiple physics steps per video frame for smoothness:

```
for _ in range(TRANSITION_FRAMES):
    for _ in range(STEPS_PER_FRAME):
        swarm.step()   # RK4 update
```

8 Task–Solution Mapping (Requirement-Oriented)

Requirement Item	Implemented Solution	Mathematical Basis	Code Re
Input formation = handwritten name	Use final state of Subtask 1	IVP initial condition	swarm.p
Extract greeting targets	Edge sampling / text rasterization	$\{T_i\} \subset \mathcal{E}$	`extract`
Assign drones to targets	Greedy nearest-neighbor matching	$D_{ij} = \lVert x_i - T_j \rVert$	`greedy_`
Move to greeting smoothly	Spring–damper dynamics	Eqs. (**??**)–(**??**)	`Swarm.s`
Avoid collisions	Repulsive force within R_{safe}	f_{rep} model	**`repulsi`**
Respect speed limits	Velocity saturation	$\min(1, v_{\max}/\lVert v_i \rVert)$	saturati
Numerical method stated/justified	RK4 integration	4th-order IVP solver	RK4 in
Provide trajectories + visualization	Save per-step positions	$x_i(t)$ samples	trajector

9 Practical Considerations and Limitations

- **Assignment optimality:** Greedy matching is efficient but not globally optimal; in rare cases it can introduce crossings. A Hungarian assignment can reduce travel cost but is more expensive for large N.
- **Parameter tuning:** Large k_p increases responsiveness but may amplify repulsion-induced stiffness; k_d must be tuned to suppress oscillations.
- **Sparse edges:** If the greeting image has too few edge pixels, sampling must upsample or use a binary fill fallback to obtain N targets.

10 Conclusion

Subtask 2 is formulated as an IVP in which the initial swarm state is inherited from the handwritten-name formation and the target set is replaced by a new static greeting formation. By recomputing assignment from the current configuration and integrating a spring–damper model with collision avoidance and velocity saturation via RK4, the swarm transitions smoothly and safely to the “Happy New Year!” figure, producing trajectories suitable for visualization and subsequent dynamic extensions.

AI Usage Disclosure

AI assistance was used for report drafting and stylistic refinement, while the mathematical model and implementation decisions were based on the project requirements and verified against the provided code.