Mathematical Modeling of Drone Kinematics:

From Single-Drone Point-to-Point Motion to Scaling Multi-Drone Formations

August 2026

Contents

1 Overview

This document develops the kinematic model of drone motion in two stages:

1. **Part I** — a single drone commanded to move from an initial position to a goal position (the point-to-point, or *go-to-goal*, problem);
2. **Part II** — n drones holding a geometric formation defined by a centroid, orientation, and *scale*, with the formation Jacobian and inverse kinematics that let the group translate, rotate, and expand or contract as a single virtual body.

Throughout, kinematics means we model *motion* (positions, velocities) and treat velocity as the control input; forces, thrust, and aerodynamics are abstracted into a lower-level controller.

2 Part I: Single Drone Moving to One Position

2.1 Kinematic model

Let the drone's position in the world frame be $\boldsymbol{p}(t) \in \mathbb{R}^3$. The simplest kinematic model is the *single integrator*:

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{u}(t), \qquad \boldsymbol{u}(t) \in \mathbb{R}^3, \tag{1}$$

where $\boldsymbol{u}$ is the commanded velocity. This abstraction is justified for multirotors because the onboard autopilot tracks velocity commands quickly relative to the timescale of trajectory-level motion.

A more faithful model retains the attitude. With rotation matrix $R \in SO(3)$ (body to world) and body angular velocity $\boldsymbol{\omega}_b$:

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \qquad \dot{R} = R\,\hat{\boldsymbol{\omega}}_b, \tag{2}$$

where $\hat{\cdot}$ maps a vector to its skew-symmetric matrix, $\hat{\boldsymbol{\omega}}\boldsymbol{a} = \boldsymbol{\omega} \times \boldsymbol{a}$. A quadrotor is *underactuated*: it produces thrust only along its body z-axis, so lateral acceleration requires tilting. In this document the formation layer outputs velocity references $\boldsymbol{u}$, and (2) is handled by the inner attitude loop.

2.2 Go-to-goal control law

Let $\boldsymbol{p}^* \in \mathbb{R}^3$ be the desired position. Define the position error

$$\boldsymbol{e}(t) = \boldsymbol{p}^* - \boldsymbol{p}(t). \tag{3}$$

The proportional go-to-goal law is

$$\boldsymbol{u}(t) = k\,\boldsymbol{e}(t) = k\big(\boldsymbol{p}^* - \boldsymbol{p}(t)\big), \qquad k > 0. \tag{4}$$

2.3 Closed-form solution and convergence

Substituting (4) into (1) gives the linear ODE

$$\dot{\boldsymbol{p}} = k(\boldsymbol{p}^* - \boldsymbol{p}) \quad \Longrightarrow \quad \boldsymbol{p}(t) = \boldsymbol{p}^* + \big(\boldsymbol{p}(0) - \boldsymbol{p}^*\big)e^{-kt}. \tag{5}$$

The error decays exponentially, $\|\boldsymbol{e}(t)\| = \|\boldsymbol{e}(0)\|\,e^{-kt}$, so the drone converges to the goal from any initial condition. Formally, the Lyapunov function $V = \frac{1}{2}\|\boldsymbol{e}\|^2$ satisfies

$$\dot{V} = \boldsymbol{e}^\top \dot{\boldsymbol{e}} = -k\,\|\boldsymbol{e}\|^2 \le 0, \tag{6}$$

proving global exponential stability of $\boldsymbol{p} = \boldsymbol{p}^*$.

2.4 Velocity saturation

Real drones have a maximum speed $v_{\max}$. A common modification caps the command while preserving its direction:

$$\boldsymbol{u} = \begin{cases} k\,\boldsymbol{e}, & k\|\boldsymbol{e}\| \le v_{\max}, \\ v_{\max}\,\dfrac{\boldsymbol{e}}{\|\boldsymbol{e}\|}, & \text{otherwise.} \end{cases} \tag{7}$$

Far from the goal the drone flies at $v_{\max}$ straight toward $\boldsymbol{p}^*$; near the goal it slows exponentially, arriving without overshoot.

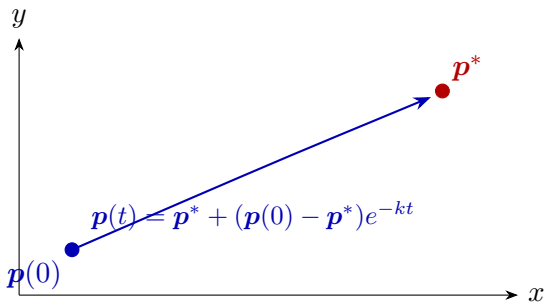

Figure 1: Exponential point-to-point trajectory of a single drone under the proportional law.

3 Part II: Multiple Drones in Formation with Scaling

3.1 Stacked multi-drone kinematics

Consider n drones with positions $\boldsymbol{p}_i \in \mathbb{R}^3$, each obeying $\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i$. Stack them:

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_n \end{bmatrix} \in \mathbb{R}^{3n}, \qquad \dot{\boldsymbol{p}} = \boldsymbol{u} \in \mathbb{R}^{3n}. \tag{8}$$

3.2 The formation as a virtual body

Define a *template* of fixed offsets $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n \in \mathbb{R}^3$ describing the desired shape (e.g., the vertices of a triangle), chosen so $\sum_i \boldsymbol{r}_i = \boldsymbol{0}$ (offsets measured from the shape's centroid). The formation state is

$$\boldsymbol{\xi} = (\boldsymbol{c},\, R,\, s), \qquad \boldsymbol{c} \in \mathbb{R}^3 \text{ (centroid)}, \quad R \in SO(3) \text{ (orientation)}, \quad s > 0 \text{ (scale)}. \tag{9}$$

3.3 Forward kinematics

Given the formation state, each drone's position is

$$\boxed{\boldsymbol{p}_i = \boldsymbol{c} + s\, R\, \boldsymbol{r}_i} \tag{10}$$

This is the forward map $\boldsymbol{p} = f(\boldsymbol{\xi})$: the analog of joint angles → end-effector pose for a robot arm. The scale s acts uniformly: doubling s doubles every drone's distance from the centroid while preserving the shape and orientation.

3.4 Differential kinematics and the formation Jacobian

Differentiate (10) using $\dot{R} = \hat{\boldsymbol{\omega}} R$:

$$\dot{\boldsymbol{p}}_i = \dot{\boldsymbol{c}} + \dot{s}\, R\boldsymbol{r}_i + s\, \hat{\boldsymbol{\omega}}\, R\boldsymbol{r}_i = \dot{\boldsymbol{c}} + \dot{s}\, R\boldsymbol{r}_i + \boldsymbol{\omega} \times (s\, R\boldsymbol{r}_i). \tag{11}$$

Writing $\boldsymbol{q}_i = R\boldsymbol{r}_i$, each drone's velocity is linear in the formation velocity $\dot{\boldsymbol{\xi}} = (\dot{\boldsymbol{c}}, \boldsymbol{\omega}, \dot{s})$:

$$\dot{\boldsymbol{p}}_i = \underbrace{\begin{bmatrix} I_3 & -s\,\hat{\boldsymbol{q}}_i & \boldsymbol{q}_i \end{bmatrix}}_{J_i(\boldsymbol{\xi}) \in \mathbb{R}^{3\times 7}} \begin{bmatrix} \dot{\boldsymbol{c}} \\ \boldsymbol{\omega} \\ \dot{s} \end{bmatrix}, \qquad \text{so} \qquad \dot{\boldsymbol{p}} = J(\boldsymbol{\xi})\,\dot{\boldsymbol{\xi}}, \quad J = \begin{bmatrix} J_1 \\ \vdots \\ J_n \end{bmatrix} \in \mathbb{R}^{3n\times 7}. \tag{12}$$

J is the *formation Jacobian*. Note the three blocks: translation passes straight through (I_3), rotation contributes $\boldsymbol{\omega}\times$ the drone's offset (the $-s\hat{\boldsymbol{q}}_i$ term), and scaling pushes each drone radially along its own offset direction ($\boldsymbol{q}_i$).

3.5 Inverse kinematics of the formation

Position-level. Given a desired formation state $\boldsymbol{\xi}_d = (\boldsymbol{c}_d, R_d, s_d)$, the desired drone positions follow directly from the forward map:

$$\boldsymbol{p}_i^* = \boldsymbol{c}_d + s_d\, R_d\, \boldsymbol{r}_i, \qquad i = 1, \dots, n. \tag{13}$$

Velocity-level. Given a desired formation trajectory $\boldsymbol{\xi}_d(t)$ with velocity $\dot{\boldsymbol{\xi}}_d = (\dot{\boldsymbol{c}}_d, \boldsymbol{\omega}_d, \dot{s}_d)$, the feedforward-plus-feedback control for drone i is

$$\boxed{\boldsymbol{u}_i = \underbrace{\dot{\boldsymbol{c}}_d + \dot{s}_d\, R_d \boldsymbol{r}_i + \boldsymbol{\omega}_d \times \big(s_d R_d \boldsymbol{r}_i\big)}_{\text{feedforward } J_i \dot{\boldsymbol{\xi}}_d} + \underbrace{k\big(\boldsymbol{p}_i^*(t) - \boldsymbol{p}_i\big)}_{\text{feedback correction}}} \tag{14}$$

Each drone's tracking error $\boldsymbol{e}_i = \boldsymbol{p}_i^* - \boldsymbol{p}_i$ again obeys $\dot{\boldsymbol{e}}_i = -k\boldsymbol{e}_i$, so *Part I's convergence result applies drone-by-drone*: the single-drone go-to-goal law is the elementary building block of formation tracking.

Redundancy. The drones have $3n$ degrees of freedom but the formation has only 7. When $3n > 7$ the map is redundant; secondary objectives (collision avoidance, energy) can be inserted in the null space of the Jacobian:

$$\boldsymbol{u} = J\dot{\boldsymbol{\xi}}_d + \big(I_{3n} - J^\dagger J\big)\, \boldsymbol{u}_{\text{sec}}, \tag{15}$$

where $J^\dagger = (J^\top J)^{-1} J^\top$ is the Moore–Penrose pseudoinverse. The projected term moves drones without changing the formation-level motion — the same trick used for redundant robot manipulators.

3.6 Pure scaling maneuver

To expand or contract the formation in place, set $\dot{\boldsymbol{c}}_d = \boldsymbol{0}$, $\boldsymbol{\omega}_d = \boldsymbol{0}$, and prescribe a scale trajectory $s_d(t)$, e.g., exponential convergence to a target scale s^*:

$$\dot{s}_d = \lambda\,(s^* - s_d), \qquad \lambda > 0 \quad \Longrightarrow \quad s_d(t) = s^* + \big(s_d(0) - s^*\big)e^{-\lambda t}. \tag{16}$$

Then (14) reduces to a purely radial command

$$\boldsymbol{u}_i = \dot{s}_d\, R_d \boldsymbol{r}_i + k\big(\boldsymbol{p}_i^* - \boldsymbol{p}_i\big), \tag{17}$$

so every drone moves along its own offset ray through the centroid, and each drone's speed is proportional to $\|\boldsymbol{r}_i\|$: outer drones move faster, preserving the shape exactly during the expansion.

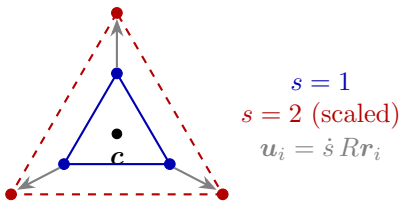

Figure 2: Pure scaling of a triangle formation about a fixed centroid. Each drone moves radially along its offset $R\boldsymbol{r}_i$; shape and orientation are preserved.

3.7 Worked example: three drones in a scaling triangle

Take $n = 3$ drones in the horizontal plane (z fixed), an equilateral-triangle template of unit circumradius,

$$\boldsymbol{r}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{r}_2 = \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}, \quad \boldsymbol{r}_3 = \begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix}, \qquad \sum_i \boldsymbol{r}_i = \boldsymbol{0}, \tag{18}$$

with planar rotation $R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$. Command the formation to hold $\boldsymbol{c}_d = (5, 5)$, $\theta_d = 0$, and expand from $s_d(0) = 1$ to $s^* = 2$ with $\dot{s}_d = \lambda(s^* - s_d)$. The desired positions at any instant are

$$\boldsymbol{p}_1^* = \begin{bmatrix} 5 + s_d \\ 5 \end{bmatrix}, \quad \boldsymbol{p}_2^* = \begin{bmatrix} 5 - \frac{s_d}{2} \\ 5 + \frac{\sqrt{3}}{2} s_d \end{bmatrix}, \quad \boldsymbol{p}_3^* = \begin{bmatrix} 5 - \frac{s_d}{2} \\ 5 - \frac{\sqrt{3}}{2} s_d \end{bmatrix}, \tag{19}$$

and the commands from (14) are $\boldsymbol{u}_i = \dot{s}_d\, \boldsymbol{r}_i + k(\boldsymbol{p}_i^* - \boldsymbol{p}_i)$. As $t \to \infty$, $s_d \to 2$: the triangle doubles in size about the fixed centroid $(5, 5)$, and the drone-level errors decay as e^{-kt} exactly as in Part I.

3.8 Decentralized formulation (scale-consensus sketch)

The controller (14) assumes each drone knows the global formation state. To decentralize, define the formation on a communication graph G with neighbor sets $\mathcal{N}_i$ and use displacement constraints with a shared scale estimate $\hat{s}_i$ per drone:

$$\boldsymbol{u}_i = -\sum_{j \in \mathcal{N}_i} \Big[(\boldsymbol{p}_i - \boldsymbol{p}_j) - \hat{s}_i\, R(\boldsymbol{r}_i - \boldsymbol{r}_j) \Big], \qquad \dot{\hat{s}}_i = -\gamma \sum_{j \in \mathcal{N}_i} (\hat{s}_i - \hat{s}_j) + (\text{leader input}). \tag{20}$$

Each drone uses only relative positions of its neighbors; the scale estimates reach consensus over the connected graph, so a single leader adjusting its $\hat{s}$ makes the entire swarm expand or contract cooperatively.

4 Summary of the Model Hierarchy

Level	Model	Key equation
Single drone	Single integrator, go-to-goal	$\boldsymbol{u} = k(\boldsymbol{p}^* - \boldsymbol{p})$
Formation (forward)	Virtual body: centroid, rotation, scale	$\boldsymbol{p}_i = \boldsymbol{c} + sR\boldsymbol{r}_i$
Formation (differential)	Formation Jacobian	$\dot{\boldsymbol{p}} = J(\boldsymbol{\xi})\dot{\boldsymbol{\xi}}$
Formation (inverse)	Feedforward + feedback tracking	Eq. (14)
Scaling	Radial motion, shape-preserving	$\boldsymbol{u}_i = \dot{s}\, R\boldsymbol{r}_i + k\boldsymbol{e}_i$
Decentralized	Graph Laplacian + scale consensus	$\boldsymbol{u}_i = -\sum_j [(\boldsymbol{p}_i - \boldsymbol{p}_j) - \hat{s}_i R(\boldsymbol{r}_i - \boldsymbol{r}_j)]$

The single-drone point-to-point law is not merely a warm-up: it is the exact feedback term reused inside the formation controller, with the formation's forward kinematics supplying each drone's moving goal $\boldsymbol{p}_i^*(t)$ and the Jacobian supplying the feedforward velocity. Scaling enters as one additional degree of freedom of the virtual body, commanded through $\dot{s}$ and realized as shape-preserving radial motion.