

PINN Physics Layer Reference

Physics Laws & Equations Used in PINN Physics Layer

1. Rotational Dynamics (Euler's Equations)

Equation 1.1 - Roll Angular Acceleration:

$$\dot{p} = \frac{J_{yy} - J_{zz}}{J_{xx}} \cdot q \cdot r + \frac{\tau_x}{J_{xx}} \quad (1)$$

where:

- p = roll rate (rad/s)
- \dot{p} = roll angular acceleration (rad/s²)
- q = pitch rate (rad/s)
- r = yaw rate (rad/s)
- τ_x = roll torque (N · m)
- J_{xx} = moment of inertia about x-axis = 6.86×10^{-5} kg · m²
- J_{yy} = moment of inertia about y-axis = 9.2×10^{-5} kg · m²
- J_{zz} = moment of inertia about z-axis = 1.366×10^{-4} kg · m²

Equation 1.2 - Pitch Angular Acceleration:

$$\dot{q} = \frac{J_{zz} - J_{xx}}{J_{yy}} \cdot p \cdot r + \frac{\tau_y}{J_{yy}} \quad (2)$$

where:

- q = pitch rate (rad/s)
- \dot{q} = pitch angular acceleration (rad/s²)
- τ_y = pitch torque (N · m)

Equation 1.3 - Yaw Angular Acceleration:

$$\dot{r} = \frac{J_{xx} - J_{yy}}{J_{zz}} \cdot p \cdot q + \frac{\tau_z}{J_{zz}} \quad (3)$$

where:

- r = yaw rate (rad/s)
- \dot{r} = yaw angular acceleration (rad/s²)
- τ_z = yaw torque (N · m)

Key Feature: Real Euler equations with NO artificial damping terms. Pure physics-based rotational dynamics.

2. Simplified Euler Angle Integration

Equation 2.1 - Roll Angle Rate:

$$\dot{\phi} = p \quad (4)$$

where:

- ϕ = roll angle (rad)
- $\dot{\phi}$ = roll angle rate (rad/s)
- p = roll rate (rad/s)

Equation 2.2 - Pitch Angle Rate:

$$\dot{\theta} = q \quad (5)$$

where:

- θ = pitch angle (rad)
- $\dot{\theta}$ = pitch angle rate (rad/s)
- q = pitch rate (rad/s)

Equation 2.3 - Yaw Angle Rate:

$$\dot{\psi} = r \quad (6)$$

where:

- ψ = yaw angle (rad)
- $\dot{\psi}$ = yaw angle rate (rad/s)
- r = yaw rate (rad/s)

Note: These are simplified angular integrations valid for small angles, not the full nonlinear Euler kinematics.

3. Translational Dynamics (Vertical Motion)

Equation 3.1 - Vertical Acceleration:

$$\dot{v}_z = -g + \frac{T}{m \cdot \cos(\theta) \cdot \cos(\phi)} \quad (7)$$

where:

- v_z = vertical velocity (m/s, positive downward in NED frame)
- \dot{v}_z = vertical acceleration (m/s²)
- T = total thrust (N)
- θ = pitch angle (rad)
- ϕ = roll angle (rad)
- m = quadrotor mass = 0.068 kg
- g = gravitational acceleration = 9.81 m/s²

Equation 3.2 - Altitude Rate:

$$\dot{z} = v_z \quad (8)$$

where:

- z = altitude (m, positive downward in NED frame)
- \dot{z} = altitude rate (m/s)
- v_z = vertical velocity (m/s)

Note: No aerodynamic drag is included in the physics loss formulation.

Equations Used in Data Generation but NOT in PINN

4. Full Nonlinear Euler Kinematics (Data Generation Only)

Equation 4.1 - Full Roll Angle Rate:

$$\dot{\phi} = p + \sin(\phi) \tan(\theta) \cdot q + \cos(\phi) \tan(\theta) \cdot r \quad (9)$$

Equation 4.2 - Full Pitch Angle Rate:

$$\dot{\theta} = \cos(\phi) \cdot q - \sin(\phi) \cdot r \quad (10)$$

Equation 4.3 - Full Yaw Angle Rate:

$$\dot{\psi} = \frac{\sin(\phi) \cdot q + \cos(\phi) \cdot r}{\cos(\theta)} \quad (11)$$

Why not in PINN: PINN uses simplified small-angle approximations ($\dot{\phi} = p$, $\dot{\theta} = q$, $\dot{\psi} = r$) which are valid for typical quadrotor maneuvers with angles $< 30^\circ$.

5. Full 6DOF Translational Dynamics (Data Generation Only)

Equation 5.1 - X-Velocity Acceleration:

$$\dot{u} = r \cdot v - q \cdot w + \frac{f_x}{m} - g \sin(\theta) - c_d \cdot u \cdot |u| \quad (12)$$

Equation 5.2 - Y-Velocity Acceleration:

$$\dot{v} = p \cdot w - r \cdot u + \frac{f_y}{m} + g \cos(\theta) \sin(\phi) - c_d \cdot v \cdot |v| \quad (13)$$

Equation 5.3 - Z-Velocity Acceleration (with drag):

$$\dot{w} = q \cdot u - p \cdot v + \frac{f_z}{m} + g \cos(\theta) \cos(\phi) - c_d \cdot w \cdot |w| \quad (14)$$

where:

- u, v, w = body-frame velocities in x, y, z directions (m/s)
- f_x, f_y, f_z = body-frame forces (N)
- $c_d = 0.05 \text{ kg/m} = \text{quadratic drag coefficient}$
- Coriolis terms: $r \cdot v - q \cdot w$, $p \cdot w - r \cdot u$, $q \cdot u - p \cdot v$

Why not in PINN: PINN models only vertical (z-axis) dynamics for altitude control. Horizontal motion (u, v) is not controlled in the training data, and the simplified model assumes $f_x = f_y = 0$. Additionally, drag forces are omitted from physics loss.

6. Body-to-World Frame Transformation (Data Generation Only)

Equation 6.1 - X-Position Rate:

$$\dot{x} = \cos(\psi) \cos(\theta) \cdot u + [\cos(\psi) \sin(\theta) \sin(\phi) - \sin(\psi) \cos(\phi)] \cdot v + [\sin(\psi) \sin(\phi) + \cos(\psi) \sin(\theta) \cos(\phi)] \cdot w \quad (15)$$

Equation 6.2 - Y-Position Rate:

$$\dot{y} = \sin(\psi) \cos(\theta) \cdot u + [\cos(\psi) \cos(\phi) + \sin(\psi) \sin(\theta) \sin(\phi)] \cdot v + [\sin(\psi) \sin(\theta) \cos(\phi) - \cos(\psi) \sin(\phi)] \cdot w \quad (16)$$

Equation 6.3 - Z-Position Rate:

$$\dot{z} = -[\sin(\theta) \cdot u - \cos(\theta) \sin(\phi) \cdot v - \cos(\theta) \cos(\phi) \cdot w] \quad (17)$$

Why not in PINN: PINN focuses on body-frame dynamics, not world-frame position tracking. Position is not required for learning the dynamics model.

7. Motor Thrust/Torque Mapping (Not Currently Used)

Equation 7.1 - Total Thrust from Motors:

$$T = k_t \cdot (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (18)$$

Equation 7.2 - Roll Torque from Motors:

$$\tau_x = k_t \cdot L \cdot (\omega_2^2 - \omega_4^2) \quad (19)$$

Equation 7.3 - Pitch Torque from Motors:

$$\tau_y = k_t \cdot L \cdot (\omega_3^2 - \omega_1^2) \quad (20)$$

Equation 7.4 - Yaw Torque from Motors:

$$\tau_z = k_q \cdot (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (21)$$

where:

- $\omega_1, \omega_2, \omega_3, \omega_4$ = motor speeds (rad/s)
- $k_t = 0.01 \text{ N}/(\text{rad/s})^2$ = thrust coefficient
- $k_q = 7.8263 \times 10^{-4} \text{ N} \cdot \text{m}/(\text{rad/s})^2$ = torque coefficient
- L = arm length from center to motor (m)

Why not in PINN: Currently, PINN receives thrust T and torques τ_x, τ_y, τ_z as direct inputs rather than computing them from motor speeds. These equations are only used to calculate physical limits during data generation.

Complete Mapping: 18 State Outputs to Equations

8. State Variables and Their Physics Equations

#	Output	Equation Used	Status	Reason/Note
Rotational Dynamics				
1	\dot{p}	Eq 1.1: $\dot{p} = \frac{J_{yy}-J_{zz}}{J_{xx}} \cdot q \cdot r + \frac{\tau_x}{J_{xx}}$	USED	Full Euler equation
2	\dot{q}	Eq 1.2: $\dot{q} = \frac{J_{zz}-J_{xx}}{J_{yy}} \cdot p \cdot r + \frac{\tau_y}{J_{yy}}$	USED	Full Euler equation
3	\dot{r}	Eq 1.3: $\dot{r} = \frac{J_{xx}-J_{yy}}{J_{zz}} \cdot p \cdot q + \frac{\tau_z}{J_{zz}}$	USED	Full Euler equation
4	p	Integration: $p_{t+dt} = p_t + \dot{p} \cdot dt$	USED	Direct integration
5	q	Integration: $q_{t+dt} = q_t + \dot{q} \cdot dt$	USED	Direct integration
6	r	Integration: $r_{t+dt} = r_t + \dot{r} \cdot dt$	USED	Direct integration
Euler Angle Kinematics				
7	$\dot{\phi}$	Eq 2.1 (PINN): $\dot{\phi} = p$ Eq 4.1 (Data): Full non-linear	SIMPLIFIED	Small-angle approximation valid for $ \phi < 30^\circ$. Data uses full kinematics with sin, tan terms.
8	$\dot{\theta}$	Eq 2.2 (PINN): $\dot{\theta} = q$ Eq 4.2 (Data): Full non-linear	SIMPLIFIED	Small-angle approximation valid for $ \theta < 30^\circ$. Data uses full kinematics with cos, sin terms.
9	$\dot{\psi}$	Eq 2.3 (PINN): $\dot{\psi} = r$ Eq 4.3 (Data): Full non-linear	SIMPLIFIED	Small-angle approximation. Data uses full kinematics with gimbal lock at $\theta = \pm 90^\circ$.
10	ϕ	Integration: $\phi_{t+dt} = \phi_t + \dot{\phi} \cdot dt$	USED	Uses simplified $\dot{\phi}$
11	θ	Integration: $\theta_{t+dt} = \theta_t + \dot{\theta} \cdot dt$	USED	Uses simplified $\dot{\theta}$
12	ψ	Integration: $\psi_{t+dt} = \psi_t + \dot{\psi} \cdot dt$	USED	Uses simplified $\dot{\psi}$
Translational Dynamics				
13	\dot{u}	Eq 5.1: Full 6DOF with Coriolis and drag	NOT USED	PINN focuses on vertical-only control. No horizontal force inputs ($f_x = 0$) in training data.

#	Output	Equation Used	Status	Reason/Note
14	\dot{v}	Eq 5.2: Full 6DOF with Coriolis and drag	NOT USED	PINN focuses on vertical-only control. No horizontal force inputs ($f_y = 0$) in training data.
15	\dot{w}	Eq 3.1 (PINN): No drag Eq 5.3 (Data): With drag	SIMPLIFIED	PINN omits Coriolis terms (assumes $u = v = 0$) and drag ($c_d \cdot w \cdot w $) for simpler vertical dynamics.
16	u	Integration of \dot{u}	NOT USED	Not predicted since \dot{u} is not modeled. Passive/uncontrolled horizontal motion.
17	v	Integration of \dot{v}	NOT USED	Not predicted since \dot{v} is not modeled. Passive/uncontrolled horizontal motion.
18	w (v_z)	Integration: $w_{t+dt} = w_t + \dot{w} \cdot dt$	USED	Uses simplified \dot{w}

Legend

- **USED** - Equation actively used in PINN physics loss (perfect match with data generation)
- **SIMPLIFIED** - PINN uses a simplified approximation; data generation uses full complex equation
- **NOT USED** - Variable exists in training data but completely absent from PINN physics loss

Detailed Explanations

Why SIMPLIFIED (4 variables):

- **Outputs 7-9 (Euler angle rates):** PINN uses small-angle approximations ($\dot{\phi} = p$, $\dot{\theta} = q$, $\dot{\psi} = r$) which are valid for typical quadrotor maneuvers with angles $< 30^\circ$. Data generation uses full nonlinear Euler kinematics with trigonometric coupling terms (\sin, \cos, \tan).
- **Output 15 (vertical acceleration):** PINN omits Coriolis coupling terms (assumes horizontal velocities $u = v = 0$) and aerodynamic drag ($c_d \cdot w \cdot |w|$) for a simpler vertical-only dynamics model.

Why NOT USED (4 variables):

- **Outputs 13-14 (horizontal accelerations):** PINN focuses on vertical-only quadrotor control. Training data has no horizontal control inputs ($f_x = f_y = 0$), making

horizontal dynamics passive/uncontrolled. Not needed for altitude and attitude stabilization task.

- **Outputs 16-17 (horizontal velocities):** Cannot be predicted since their derivatives (\dot{u}, \dot{v}) are not modeled in PINN. These are byproducts of the full 6DOF simulation but not required for the vertical control objective.

Summary of PINN Model Scope

What PINN Models:

- Full 3D rotational dynamics (Euler's equations)
- Simplified Euler angle integration (small-angle approximation)
- Vertical translational dynamics (without drag)
- Altitude tracking

What PINN Does NOT Model:

- Full nonlinear Euler kinematics
- Horizontal motion (x, y velocities and positions)
- Aerodynamic drag forces
- Body-to-world frame transformations
- Motor speed to thrust/torque mapping

Justification: The PINN focuses on a simplified vertical flight dynamics model suitable for altitude control and small-angle attitude stabilization, which covers the majority of typical quadrotor flight scenarios.