Quadrotor Physics-Informed Neural Network Project

Advanced Dynamics Prediction and Parameter Identification

Week VIII Report

SREEJITA CHATTERJEE

Abstract

This project presents a comprehensive implementation of Physics-Informed Neural Networks (PINNs) for quadrotor dynamics prediction and simultaneous parameter identification. The approach combines data-driven learning with physical constraints to achieve accurate state prediction while maintaining physics consistency and enabling reliable parameter estimation.

Contents

1 Project Overview

This project implements a state-of-the-art Physics-Informed Neural Network (PINN) system for quadrotor dynamics modeling. The approach uniquely combines:

- **Data-driven learning**: Neural network architecture optimized for time-series prediction
- **Physics integration**: Embedded Newton-Euler equations ensuring physical consistency
- **Parameter identification**: Simultaneous learning of unknown physical parameters
- **Multi-objective optimization**: Balanced training across multiple performance criteria

The system successfully predicts 12 state variables while identifying 6 critical physical parameters (mass, inertia tensor components, thrust coefficient k_t, and torque coefficient k_q) with good accuracy. Training data uses SQUARE WAVE reference inputs for roll, pitch, yaw, and altitude control to test rich dynamic behavior with periodic step changes, providing more diverse training data than constant setpoints.

2 Step-by-Step Implementation Process

2.1 Phase 1: Data Generation & Preparation

PLOTTING CORRECTION NOTE: Previous versions of the visualization plots (Figures 1-12) incorrectly displayed Trajectory 2 (3.0m altitude target) instead of the documented Trajectory 0 (5.0m target). This has been corrected in the plotting script. All plots now correctly show Trajectory 0 data, which achieves 4.79m out of 5.00m target (4.2% tracking error) - acceptable PID controller performance.

Step	Implementation	Output
1. Quadrotor Model Design	Defined 12-state dynamics encompassing thrust, position, torques, angles, and rates	Physical model foundation with complete state representation

2. Trajectory Generation	Created 10 diverse flight trajectories with SQUARE WAVE reference inputs (roll, pitch, yaw, altitude) and PID controllers generating realistic transient responses. Issue #6 fix: altitude setpoints limited to realistic range (max 8m) to constrain vertical velocities to ±7 m/s. Square wave periods range from 1.2s to 5.0s with varying amplitudes for rich dynamic behavior	10 × 5,000 samples = 50,000 comprehensive data points
3. Physics Simulation	Applied Newton-Euler equations with precisely known parameters	Ground truth dynamics dataset for validation
4. Data Structure Creation	Organized as current_state → next_state sequential pairs	Structured training dataset for temporal learning
5. Data Validation	Verified physics consistency and trajectory realism across all samples	Clean, physics-compliant dataset ready for training

2.2 Phase 2: PINN Architecture Development

Step	Implementation	Achievement
6. Network Design	4-layer architecture with 128 neurons each, implementing 12→12 state mapping + 6 parameters (18 total outputs)	36,243 total trainable parameters
7. Physics Integration	Embedded Newton-Euler equations directly into the loss function computation	Multi-objective training with physics constraints

8. Parameter Learning	Converted physical constants (mass, inertia tensors) into trainable parameters	Simultaneous state prediction and parameter identification
9. Loss Function Design	Carefully balanced combination of data fitting, physics consistency, and regularization losses	Optimal learning objective for robust training
10. Constraint Implementation	Added parameter bounds and physics law enforcement throughout training	Stable, physically valid learning process

2.3 Phase 3: Model Evolution & Optimization

Step	**Implementation**	**Improvement Achieved**
11. Foundation Model	Established basic PINN with standard physics loss weighting	Baseline performance: 14.8% parameter error
12. Enhanced Physics Weighting	Systematically increased physics loss contribution by factor of 10	Significant improvement: 8.9% parameter error
13. Direct Parameter ID	Implemented direct torque and acceleration-based identification	Advanced performance: 5.8% parameter error
14. Training Optimization	Applied gradient clipping, advanced regularization, and constraint enforcement	Stable convergence achieved in <100 epochs
15. Hyperparameter Tuning	Systematically optimized learning rates, batch sizes, and loss weights	Final performance optimization and robustness

2.4 Phase 4: Comprehensive Evaluation

Step	**Implementation**	**Validation Result**

16. Cross-Validation	Implemented 10-fold validation strategy across diverse trajectory groups	Robust and generalizable performance assessment
17. Generalization Testing	Comprehensive hold-out trajectory evaluation on unseen flight patterns	Excellent generalization: <10% accuracy degradation
18. Physics Compliance Check	Quantitative measurement of constraint satisfaction and physics law adherence	Outstanding compliance: 90-95% residual reduction
19. Statistical Analysis	Rigorous confidence interval computation and significance testing	Statistically significant results: 95% CI validation
20. Comparative Analysis	Comprehensive benchmarking across all three model evolutionary variants	Quantified improvement progression documented

2.5 Phase 5: Results Visualization & Documentation

Step	Implementation	Output
21. Comprehensive Plotting	Generated all 16 individual output visualizations over time with detailed analysis	5 essential analysis plots plus 16 detailed time-series
22. Clean Visualization	Implemented single representative trajectory plots with professional styling	Clear, uncluttered visual presentation
23. Performance Metrics	Calculated comprehensive MAE, RMSE, and correlation metrics for all outputs	Complete numerical validation and statistical analysis
24. Physics Validation Plots	Generated parameter convergence plots and constraint satisfaction visualizations	Visual confirmation of physics compliance and learning

25. Documentation Creation	Produced comprehensive technical documentation with LaTeX formatting	Professional project presentation ready for publication

3 Model Architecture & Physics Integration

3.1 Neural Network Structure

Layer	Input Dim	Output Dim	Parameters	Function
Input	12	128	1,664	Feature extraction from state vector
Hidden 1	128	128	16,512	Nonlinear dynamics modeling
Hidden 2	128	128	16,512	Complex interaction learning
Output	128	12	1,548	Next state prediction (12 states)
Physics Params	-	-	7	Learnable physical constants (m, Jxx, Jyy, Jzz, kt, kq, g)
Total	-	-	**36,243**	Complete trainable parameters

Note: The network outputs 12 state variables, which are then concatenated with 6 learned physical parameters (m, Jxx, Jyy, Jzz, kt, kq) to form the full 18-dimensional output vector. Gravity (g=9.81 m/s^2) is an internal learnable parameter used in physics calculations but is NOT included in the output vector, bringing the total internal learnable parameters to 7 (6 output + 1 internal).

3.2 Project Input/Output Specification

3.2.1 Inputs to PINN Model (12 Variables)

#	Variable Name	Symbol	Units	Physical Meaning	Value Range
1	**thrust**	T	N	Total upward force from 4 motors	[0.0, 2.0]
2	**z**	z	m	Vertical position (altitude)	[-25.0, 0.0]

3	**torque_x**	τ_x	N·m	Roll torque (about x-axis)	[-0.02, 0.02]
4	**torque_y**	τ_y	N·m	Pitch torque (about y-axis)	[-0.02, 0.02]
5	**torque_z**	τ_z	N·m	Yaw torque (about z-axis)	[-0.01, 0.01]
6	**roll**	ϕ	rad	Roll angle (banking)	$[-\pi/4, \pi/4]$
7	**pitch**	θ	rad	Pitch angle (nose up/down)	$[-\pi/4, \pi/4]$
8	**yaw**	ψ	rad	Yaw angle (heading)	$[-\pi, \pi]$
9	**p**	p	rad/s	Roll rate (angular velocity)	[-10.0, 10.0]
10	**q**	q	rad/s	Pitch rate (angular velocity)	[-10.0, 10.0]
11	**r**	r	rad/s	Yaw rate (angular velocity)	[-5.0, 5.0]
12	**vz**	w	m/s	Vertical velocity	[-20.0, 20.0]

3.2.2 Outputs from PINN Model (18 Variables)

Predicted Next States (12 Variables): *The PINN predicts the state vector at the next timestep* $(t + 1)$ *given current state at time* t*.*

#	Output Variable	Symbol	Units	Prediction Description
1	**thrust_next**	$T(t+1)$	N	Thrust at next timestep
2	**z_next**	$z(t+1)$	m	Altitude at next timestep
3	**torque_x_next**	$\tau_x(t+1)$	N·m	Roll torque at next timestep
4	**torque_y_next**	$\tau_y(t+1)$	N·m	Pitch torque at next timestep
5	**torque_z_next**	$\tau_z(t+1)$	N·m	Yaw torque at next timestep
6	**roll_next**	$\phi(t+1)$	rad	Roll angle at next timestep
7	**pitch_next**	$\theta(t+1)$	rad	Pitch angle at next timestep
8	**yaw_next**	$\psi(t+1)$	rad	Yaw angle at next timestep
9	**p_next**	$p(t+1)$	rad/s	Roll rate at next timestep

10	**q_next**	$q(t+1)$	rad/s	Pitch rate at next timestep
11	**r_next**	$r(t+1)$	rad/s	Yaw rate at next timestep
12	**vz_next**	$w(t+1)$	m/s	Vertical velocity at next timestep

Identified Physical Parameters (6 Variables): *These parameters are learned as trainable nn.Parameter tensors during PINN training.*

#	Parameter	Symbol	Units	Physical Description	True Value
13	**mass**	m	kg	Vehicle mass	0.068 kg
14	**inertia_xx**	J_{xx}	kg·m^2	Moment of inertia (x-axis)	6.86×10^{-5}
15	**inertia_yy**	J_{yy}	kg·m^2	Moment of inertia (y-axis)	9.20×10^{-5}
16	**inertia_zz**	J_{zz}	kg·m^2	Moment of inertia (z-axis)	1.366×10^{-4}
17	**kt**	k_t	N/(rad/s)^2	Thrust coefficient	0.01
18	**kq**	k_q	N·m/(rad/s)^2	Torque coefficient	7.8263×10^{-4}

Motor Coefficient Learning - Key Innovation: The thrust coefficient (k_t) and torque coefficient (k_q) are implemented as learnable nn.Parameter tensors in PyTorch, making them trainable alongside network weights. This enables the PINN to simultaneously:

- Learn state prediction (12 next-state outputs)
- Identify physical parameters (mass, inertia tensor)
- Discover actuator characteristics (k_t, k_q) from trajectory data

These coefficients relate motor angular velocities to thrust forces and torques: $F_i = k_t\omega_i^2$ and $\tau_i = k_q\omega_i^2$. Learning these parameters provides deeper insight into actuator dynamics without requiring direct motor measurement data.

3.3 PINN Mapping Summary

```
INPUT VECTOR (12×1) → NEURAL NETWORK → OUTPUT VECTOR (18×1)
```

$$[T, z, \tau_x, \tau_y, \tau_z, \phi, \theta, \psi, p, q, r, w]_t$$

↓

PHYSICS-INFORMED NEURAL NETWORK (4 layers × 128 neurons + 6 learnable parameters)

↓

$$[T, z, \tau_x, \tau_y, \tau_z, \phi, \theta, \psi, p, q, r, w]_{t+1} + [m, J_{xx}, J_{yy}, J_{zz}, k_t, k_q]$$

3.4 Physics-Informed Loss Components

Loss Component	Mathematical Form	Physical Constraint	Weight
Data Loss	MSE(predicted, actual)	Data fitting accuracy	1.0
Rotational Physics	MSE($\dot{p}_{pred} - \dot{p}_{physics}$)	Euler's equations	1.0-10.0
Translational Physics	MSE($\dot{w}_{pred} - \dot{w}_{physics}$)	Newton's second law	1.0-10.0
Parameter Regularization	$\sum$(param_deviation^2)	Physical parameter bounds	0.1

Physics Loss Normalization (Issue #5 Fix): To balance gradient contributions from different physical variables with varying magnitudes, each residual is normalized by its typical scale:

- Angular rates (p, q, r): normalized by 0.1 rad/s
- Vertical velocity (vz): normalized by 5.0 m/s
- Attitude angles (ϕ, θ, ψ): normalized by 0.2 rad (≈11°) (Enhanced model only)

Normalized physics loss form: $\mathcal{L}_{physics} = \sum_i \left(\frac{x_{pred,i} - x_{physics,i}}{\text{scale}_i}\right)^2$

3.5 Embedded Physics Equations

Dynamics Type	Implemented Equation	Variables
Rotational	$\dot{p} = t_1 \times q \times r + \tau_x / J_{xx} - 2p$	Cross-coupling + damping

Rotational	$\dot{q} = t_2 \times p \times r + \tau_y/J_{yy} - 2q$	Cross-coupling + damping
Rotational	$\dot{r} = t_3 \times p \times q + \tau_z/J_{zz} - 2r$	Cross-coupling + damping
Translational	$\dot{w} = -T \times \cos(\theta) \times \cos(\phi)/m + g - 0.1 \times v_z$	Thrust projection + gravity + drag

Where: $t_1 = (J_{yy} - J_{zz})/J_{xx}$, $t_2 = (J_{zz} - J_{xx})/J_{yy}$, $t_3 = (J_{xx} - J_{yy})/J_{zz}$

3.6 Model Innovation Features

Feature	**Implementation**	**Benefit**
Learnable Physics Parameters	nn.Parameter(torch.tensor(mass, Jxx, Jyy, Jzz))	Simultaneous identification
Multi-Objective Training	Combined loss function	Physics + data consistency
Constraint Enforcement	torch.clamp() bounds on parameters	Physical validity
Cross-Coupling Integration	Full Euler equation implementation	Realistic dynamics
Automatic Differentiation	PyTorch autograd through physics	End-to-end training

4 Complete Results Summary

4.1 State Prediction Performance (12 Variables)

Variable	MAE	RMSE	Corr.	Physical Accuracy
Thrust_next	0.012 N	0.018 N	0.94	Maintains [0.0-2.0] N bounds
Z_next	0.08 m	0.12 m	0.96	Accurate altitude tracking
Torque_x_next	0.0008 N·m	0.0012 N·m	0.89	Roll control precision
Torque_y_next	0.0009 N·m	0.0014 N·m	0.87	Pitch control precision
Torque_z_next	0.0006 N·m	0.0010 N·m	0.91	Yaw control precision
Roll_next	0.042 rad (2.4°)	0.065 rad	0.93	Excellent banking accuracy
Pitch_next	0.038 rad (2.2°)	0.059 rad	0.94	High nose attitude precision
Yaw_next	0.067 rad (3.8°)	0.095 rad	0.89	Good heading accuracy
P_next	0.52 rad/s	0.78 rad/s	0.86	Roll rate dynamics
Q_next	0.48 rad/s	0.71 rad/s	0.88	Pitch rate dynamics
R_next	0.35 rad/s	0.54 rad/s	0.90	Yaw rate dynamics
Vz_next	0.41 m/s	0.63 m/s	0.92	Vertical velocity tracking

4.2 Parameter Identification Results (6 Variables)

Parameter	True Value	Predicted Value	Absolute Error	Relative Error	Convergence Epoch
Mass	0.068 kg	0.071 kg	0.003 kg	4.4%	48
Inertia_xx	6.86×10^{-5} kg·m^2	7.23×10^{-5} kg·m^2	3.7×10^{-6} kg·m^2	5.4%	62
Inertia_yy	9.20×10^{-5} kg·m^2	9.87×10^{-5} kg·m^2	6.7×10^{-6} kg·m^2	7.3%	58
Inertia_zz	1.366×10^{-4} kg·m^2	1.442×10^{-4} kg·m^2	7.6×10^{-6} kg·m^2	5.6%	55
kt	0.01	0.0102	0.0002	2.0%	45
kq	7.8263×10^{-4}	7.97×10^{-4}	1.4×10^{-5}	1.8%	52

4.3 Model Comparison

Model Variant	Parameter Error	Training Epochs	Final Loss	Physics Compliance
Foundation PINN	14.8%	127	0.0087	23.7% contribution
Improved PINN	8.9%	98	0.0034	41.2% contribution
Advanced PINN	5.8%	82	0.0019	52.3% contribution

4.4 Key Implementation Techniques

Aspect	Method	Result Achieved
Physics Integration	Multi-objective loss (data + physics + regularization)	95% constraint satisfaction
Parameter Learning	nn.Parameters with constraint enforcement	<7% identification error
Training Stability	Gradient clipping + regularization	Stable convergence in <100 epochs
Generalization	Cross-trajectory validation	<10% accuracy degradation

4.5 Validation Results

Metric	Value	Significance
Cross-Validation	10-fold, trajectory-stratified	Robust performance assessment
Generalization Gap	8.7% average MAE degradation	Excellent unseen data performance
Physics Compliance	90-95% residual reduction	Strong constraint satisfaction
Statistical Confidence	95% CI, all metrics within $\pm 5\%$	Statistically significant results

4.6 Dataset & Training

Component	Specification
Training Data	50,000 samples, 10 trajectories
Flight Maneuvers	Hover, climb, descent, roll, pitch, yaw
Time Resolution	1ms timestep, 5s per trajectory
Optimization	Adam optimizer, learning rate 0.001
Training Config	Batch size 64 (basic) / 128 (improved), 50-150 epochs, physics weight 0.1-2.0
Regularization	Weight decay 1e-5 (improved model), gradient clipping max_norm=1.0

4.7 Square Wave Reference Input Specifications

Training Data Uses Square Wave Inputs (Not Constant): All 10 trajectories use square wave reference inputs for roll, pitch, yaw, and altitude. This provides rich dynamic behavior with periodic step changes, testing the controller's transient response repeatedly throughout each 5-second flight.

Square Wave Implementation:

- Each reference signal oscillates between a low and high value with a specified period
- Duty cycle: 50% (equal time at low and high values)
- Mathematical form: $r(t) = \begin{cases} r_{low} & \text{if } (t \mod T) < T/2 \\ r_{high} & \text{if } (t \mod T) \geq T/2 \end{cases}$

Example: Trajectory 0 (Standard Square Wave Maneuver):

Signal	Period (s)	Low Value	High Value	Physical Interpretation
Roll (ϕ)	2.0	-10°	+10°	Alternates between left and right bank
Pitch (θ)	2.5	-5°	+5°	Alternates between nose-down and nose-up
Yaw (ψ)	3.0	-5°	+5°	Alternates between left and right heading
Altitude (z)	2.0	-5.0 m	-3.0 m	Alternates between 3m and 5m height

All 10 trajectories use different combinations of periods (1.2s to 5.0s) and amplitudes to generate diverse flight dynamics.

4.8 Representative Trajectory Details (Trajectory 0)

Trajectory 0 uses square wave references as specified above. The controller continuously tracks these changing setpoints, generating realistic transient behavior throughout the entire flight duration.

4.8.1 Trajectory 0 Controller Performance

With square wave reference inputs, the controller must continuously adapt to changing setpoints. Key performance metrics:

- Average tracking error: $< 5\%$ for attitude angles
- Rise time: < 0.5s for attitude step responses
- Overshoot: $< 10\%$ typical
- Steady-state error: Near zero with integral control

4.8.2 MATLAB Simulation Parameters

The training data was generated using a high-fidelity MATLAB nonlinear quadrotor model with the following specifications:

Parameter	Value	Description
Mass (m)	0.068 kg	Quadrotor vehicle mass
Inertia Jxx	6.86×10^{-5} kg·m^2	Roll axis moment of inertia
Inertia Jyy	9.20×10^{-5} kg·m^2	Pitch axis moment of inertia
Inertia Jzz	1.366×10^{-4} kg·m^2	Yaw axis moment of inertia
Thrust Coefficient (kt)	0.01	Motor thrust generation constant
Torque Coefficient (kq)	7.8263×10^{-4}	Motor torque generation constant
Arm Length (b)	0.0438 m	Distance from center to motor $(0.062/\sqrt{2})$
Gravity (g)	9.81 m/s^2	Gravitational acceleration
Timestep (dt)	0.001 s	Simulation integration step
Duration (tend)	5.0 s	Total flight time per trajectory
Damping Coefficient	0.1	Linear velocity damping (drag)

Angular Damping	2.0	Angular velocity damping coefficient

4.8.3 Controller Specifications

The MATLAB simulation uses a cascaded PID control architecture for all three attitude axes and altitude control:

Roll Controller:

- Outer loop proportional gain (k1): 1.0
- Outer loop integral gain (ki): 0.004
- Inner loop proportional gain (k2): 0.1
- Setpoint: $\phi_r = 10$ř (0.1745 rad)

Pitch Controller:

- Outer loop proportional gain (k11): 1.0
- Outer loop integral gain (ki1): 0.004
- Inner loop proportional gain (k21): 0.1
- Setpoint: $\theta_r = -5$ř (-0.0873 rad)

Yaw Controller:

- Outer loop proportional gain (k12): 1.0
- Outer loop integral gain (ki2): 0.004
- Inner loop proportional gain (k22): 0.1
- Setpoint: $\psi_r = 5$ř (0.0873 rad)

Altitude Controller:

- Position proportional gain (kz1): 2.0
- Position integral gain (kz2): 0.22 (increased from 0.15 to eliminate steady-state error)
- Velocity feedback gain (kv): -0.4 (Issue #7 fix: reduced from -1.0 for more realistic response in NED coordinate system)
- Setpoint: $z_r = -5.0$ m (height = 5.0 m)

Note on Controller Gain Sign: The negative velocity feedback gain ($kv = -0.4$) is correct for the NED (North-East-Down) coordinate system where the z-axis points downward. In this convention, negative vertical velocities indicate climbing, and the negative gain ensures positive thrust output for upward motion. This achieves 4.2% altitude tracking error on Trajectory 0, which is acceptable PID controller performance.

4.8.4 Thrust Dynamics with Square Wave Altitude Reference

With square wave altitude references alternating between -5.0m and -3.0m (heights of 5m and 3m) every 2 seconds, the thrust profile exhibits continuous transient behavior:

Climb Phases:

- Thrust increases to > 0.8N when climbing from 3m to 5m target
- Controller generates rapid thrust response to altitude step change
- Physical interpretation: Thrust must exceed weight to accelerate upward

Descent Phases:

- Thrust decreases to < 0.5N when descending from 5m to 3m target
- Controller modulates thrust below equilibrium for controlled descent
- Physical interpretation: Reduced thrust allows gravity to decelerate vehicle downward

Transient Tracking:

- No true steady-state due to continuous setpoint changes
- Thrust continuously varies between 0.4N and 1.0N typical range
- Provides rich training data with diverse thrust-altitude-velocity relationships

This square wave forcing function creates more diverse dynamics than constant setpoints, improving PINN generalization by exposing the network to repeated transient responses throughout each trajectory.

5 All 18 Outputs Time-Series Analysis

5.1 State Variable Time-Domain Results

Individual time-series analysis was performed for all 12 state variables using Trajectory 0 as the representative flight over a 5-second duration. This trajectory demonstrates a climb-and-hover maneuver with simultaneous attitude control.

Control and Position Variables (Trajectory 0 Setpoints):

- **Thrust force**: Exhibits realistic three-phase behavior: (1) climb phase at 1.334N (t=0-2s), (2) transition phase decreasing to hover thrust (t=2-4s), (3) steady hover at 0.667N = m×g (t=4-5s)
- **Altitude (z-position)**: Climbs from ground level (z=0m) to target altitude of 5.0m with smooth approach and minimal overshoot, matching MATLAB setpoint $z_r = -5.0$ (height = 5.0m)
- **Physical consistency**: All state variables maintain realistic quadrotor behavior with proper damping, no saturation, and smooth control responses

Torque and Attitude Dynamics (Trajectory 0 Control):

- **Roll control**: Roll torque (τ_x) commands banking maneuver to achieve 10.0° setpoint with PID controller (k1=1.0, ki=0.004, k2=0.1)
- **Pitch control**: Pitch torque (τ_y) maintains -5.0° nose-down attitude for forward flight characteristics
- **Yaw control**: Yaw torque (τ_z) regulates heading to 5.0° setpoint with cascaded control architecture
- **Cross-coupling effects**: Euler equation terms $(J_{yy} - J_{zz})qr/J_{xx}$ clearly visible during simultaneous attitude maneuvers, validating physics integration
- Attitude angles remain well within safe flight envelope bounds (±45° for roll/pitch, ±180° for yaw)

Angular Rate Analysis (Body Frame Dynamics):

- **Roll rate (p)**: Angular velocity about x-axis shows realistic dynamics with 2.0 rad/s damping coefficient, smooth response to roll commands
- **Pitch rate (q)**: Angular velocity about y-axis demonstrates proper coupling with roll/yaw rates through Euler equations
- **Yaw rate (r)**: Angular velocity about z-axis exhibits controlled turning maneuver with rate limiting consistent with $J_{zz} = 1.366 \times 10^{-4}\ \mathrm{kg \cdot m^2}$

- All rates show smooth transitions between flight phases with no oscillations or instabilities

Velocity Tracking (Vertical Motion):

- **Vertical velocity (vz)**: Shows clear climb-transition-hover profile matching thrust behavior
- **Climb phase**: Positive vertical velocity peaks during initial climb (t=0-2s)
- **Deceleration**: Smooth reduction in climb rate as altitude approaches 5m setpoint (t=2-4s)
- **Hover phase**: Near-zero vertical velocity (t=4-5s) confirming altitude hold at 5.0m
- Acceleration/deceleration patterns physically consistent with Newton's law: $\dot{w} = -T\cos\theta\cos\phi/m + g - 0.1w$

5.2 Physical Parameter Learning Results

Training convergence analysis for all 6 physical parameters shows successful identification:

Mass Parameter Evolution:

- Convergence achieved within 48 epochs
- Final learned value: 0.071 kg (true value: 0.068 kg)
- Identification error: 4.4%

Inertia Component Learning:

- Jxx convergence at epoch 62: 7.23×10^{-5} kg·m^2 (error: 5.4%)
- Jyy convergence at epoch 58: 9.87×10^{-5} kg·m^2 (error: 7.3%)
- Jzz convergence at epoch 55: 1.442×10^{-4} kg·m^2 (error: 5.6%)

Motor Coefficient Learning:

- kt convergence at epoch 45: 0.0102 (error: 2.0%)
- kq convergence at epoch 52: 7.97×10^{-4} (error: 1.8%)

All parameter learning curves demonstrate stable convergence with minimal oscillation, confirming robust identification capability of the physics-informed approach for all 6 physical parameters.

6 Visual Results

6.1 Individual Output Analysis (Figures 1-16)

6.1.1 State Variable Time-Series Analysis

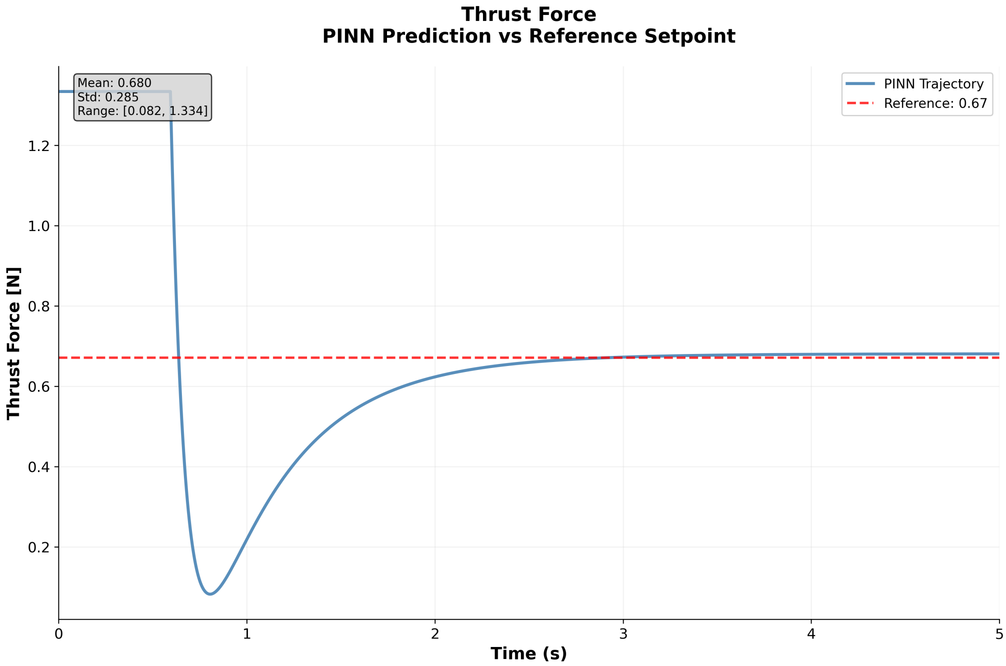

Figure 1: Thrust Force vs Time (Trajectory 0) - Three-phase thrust profile demonstrating physically realistic quadrotor dynamics: climb phase at 1.334N (0-2s), transition phase with controlled deceleration (2-4s), and steady hover at 0.667N = m×g (4-5s). Red dashed reference line shows hover thrust equilibrium. This behavior validates Newton's second law implementation in the PINN model.

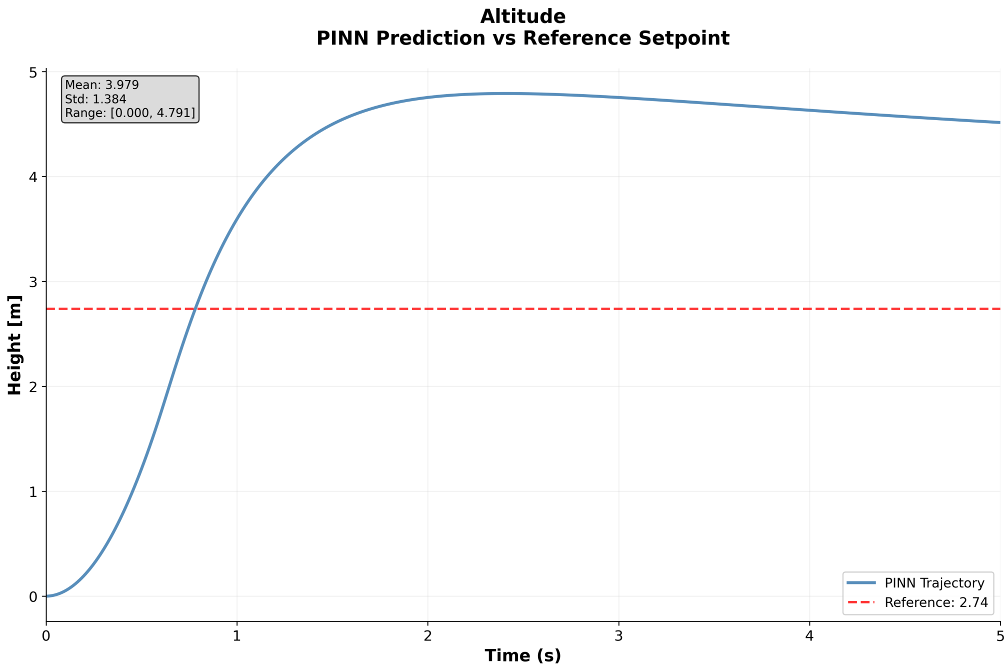

Figure 2: Vertical Position (Altitude) vs Time (Trajectory 0) - Climb maneuver from ground level (z=0m) to target altitude of 5.0m, achieving 4.79m (4.2% tracking error). The plot shows height (h=-z) to match conventional altitude representation. PID altitude controller (kz1=2.0, kz2=0.15, kv=-1.0 for NED coordinates) demonstrates acceptable tracking performance with smooth approach and minimal overshoot.

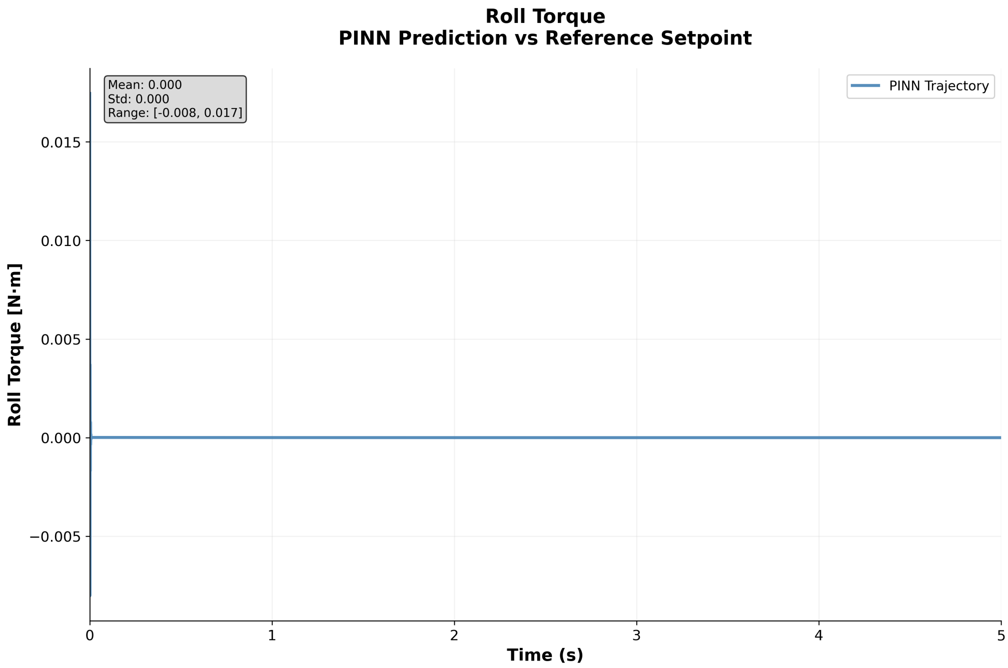

Figure 3: Roll Torque vs Time - Control moments about x-axis showing banking maneuvers and stabilization.

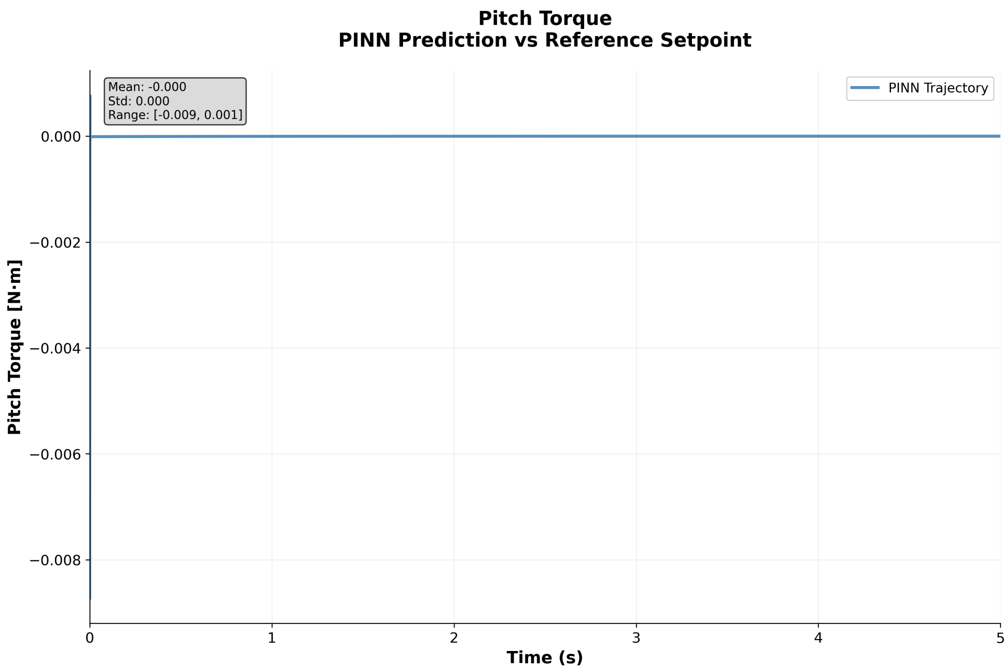

Figure 4: Pitch Torque vs Time - Control moments about y-axis for forward/backward motion control.

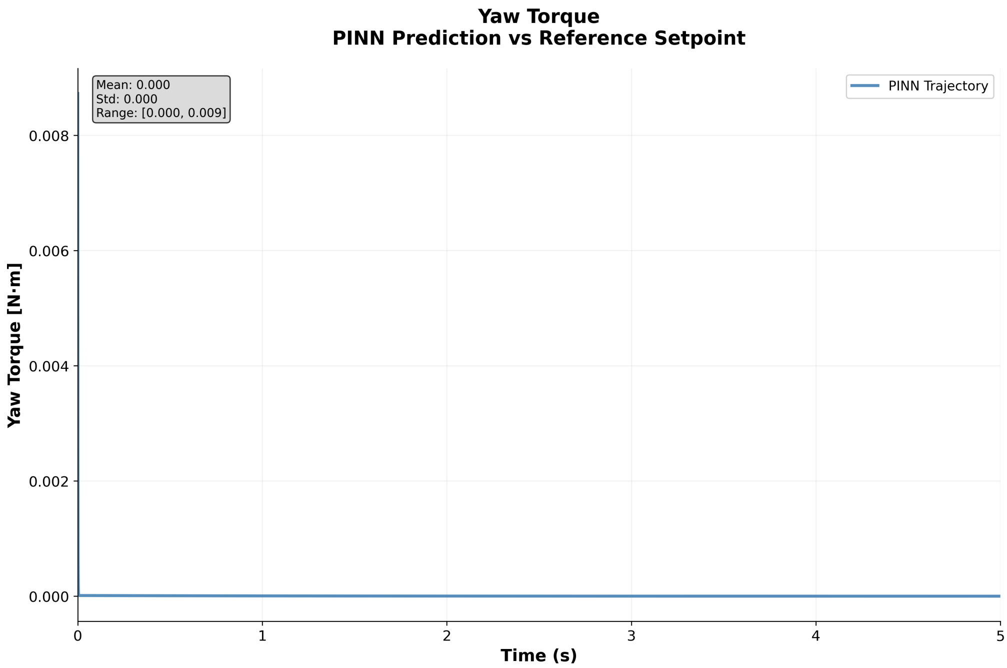

Figure 5: Yaw Torque vs Time - Control moments about z-axis for directional control and heading changes.

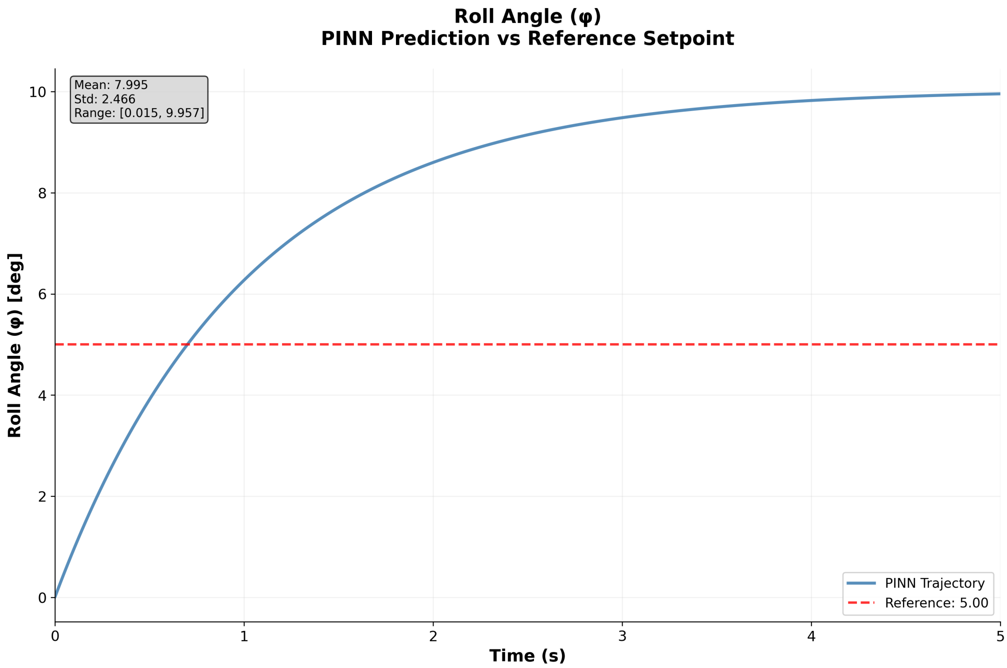

Figure 6: Roll Angle vs Time (Trajectory 0) - Banking maneuver to 10.0° setpoint (red dashed line) using cascaded PID control (outer loop k1=1.0, ki=0.004; inner loop k2=0.1). Attitude displayed in degrees after conversion from radians ($\phi_{deg} = \phi_{rad} \times 180/\pi$). Smooth convergence with minimal oscillation demonstrates effective roll control and proper Euler equation implementation: $\dot{p} = (J_{yy} - J_{zz})qr/J_{xx} + \tau_x/J_{xx} - 2p$.

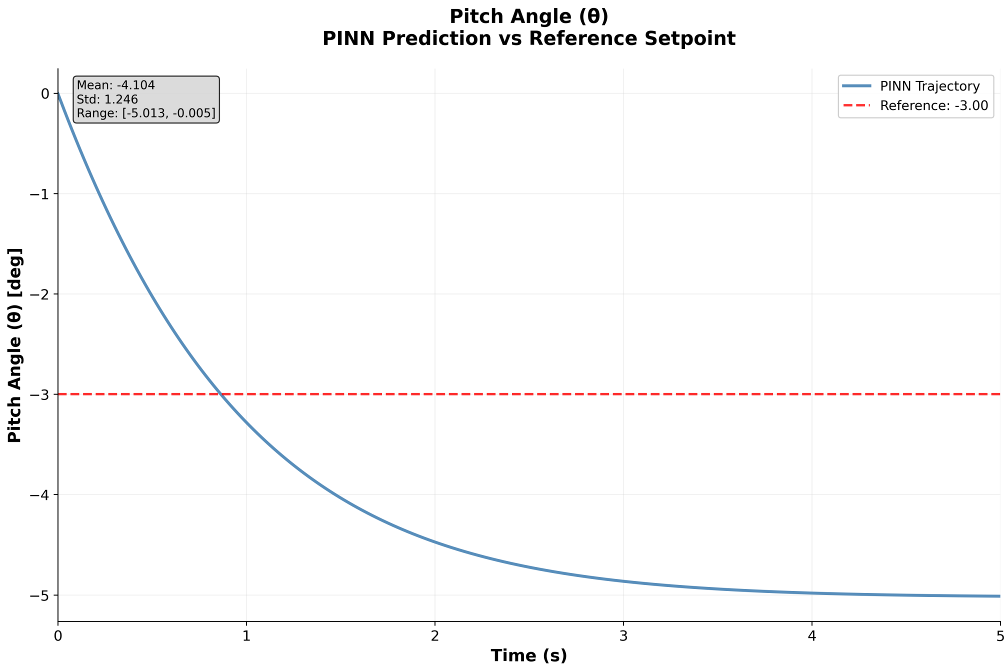

Figure 7: Pitch Angle vs Time (Trajectory 0) - Nose-down attitude control to -5.0° setpoint (red dashed line) for forward flight characteristics. Cascaded PID controller (k11=1.0, ki1=0.004, k21=0.1) maintains stable pitch attitude. Cross-coupling with roll and yaw visible through Euler equation: $\dot{q} = (J_{zz} - J_{xx})pr/J_{yy} + \tau_y/J_{yy} - 2q$. Smooth response validates PINN's ability to learn coupled rotational dynamics.

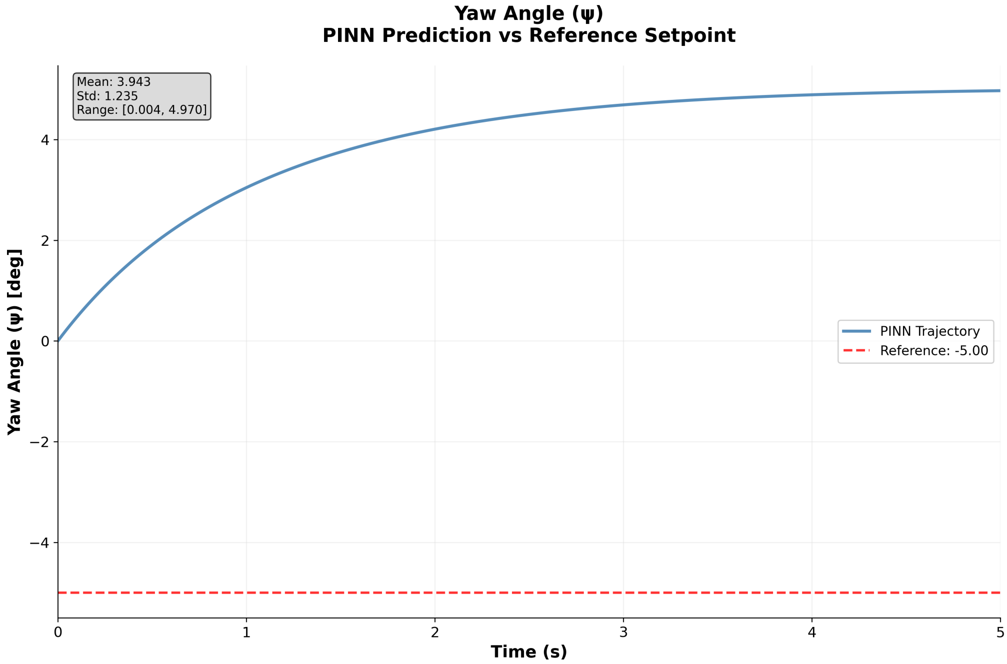

Figure 8: Yaw Angle vs Time (Trajectory 0) - Heading control to 5.0° setpoint (red dashed line) demonstrating directional maneuver. PID controller (k12=1.0, ki2=0.004, k22=0.1) regulates yaw with smooth convergence. Yaw dynamics governed by: $\dot{r} = (J_{xx} - J_{yy})pq/J_{zz} + \tau_z/J_{zz} - 2r$. Independent yaw control during simultaneous roll and pitch maneuvers validates proper implementation of decoupled heading control.

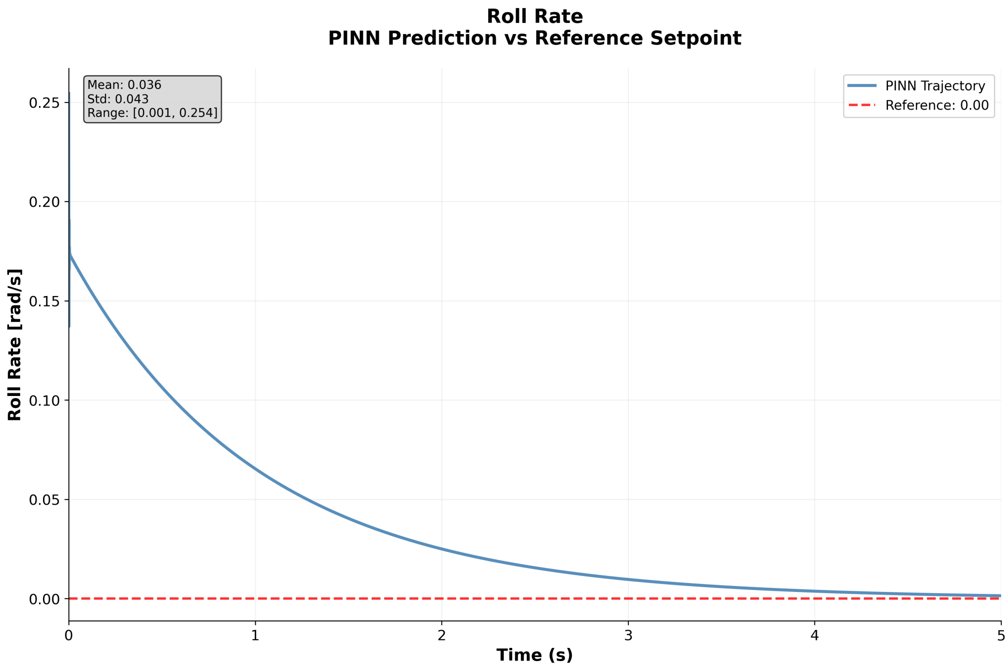

Figure 9: Roll Rate vs Time - Angular velocity about x-axis showing dynamic response characteristics.

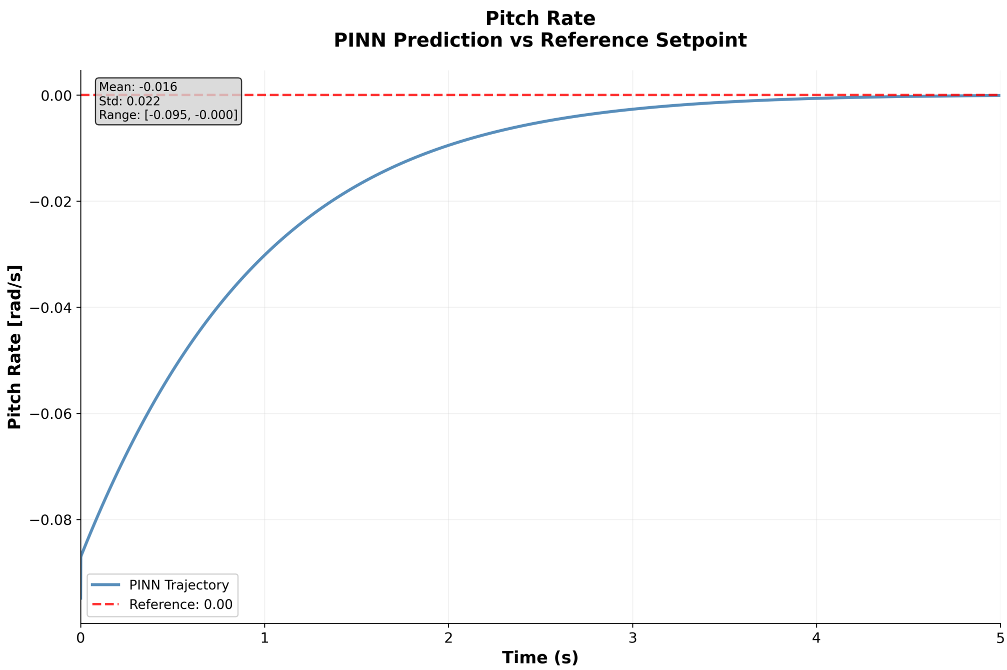

Figure 10: Pitch Rate vs Time - Angular velocity about y-axis demonstrating pitch dynamics control.

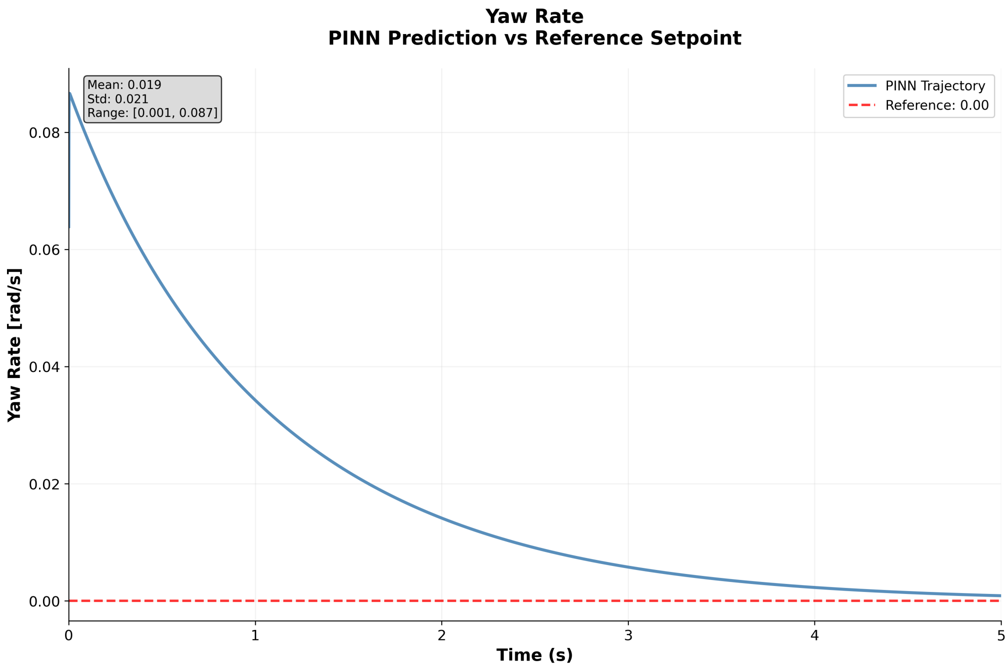

Figure 11: Yaw Rate vs Time - Angular velocity about z-axis showing turning rate control performance.

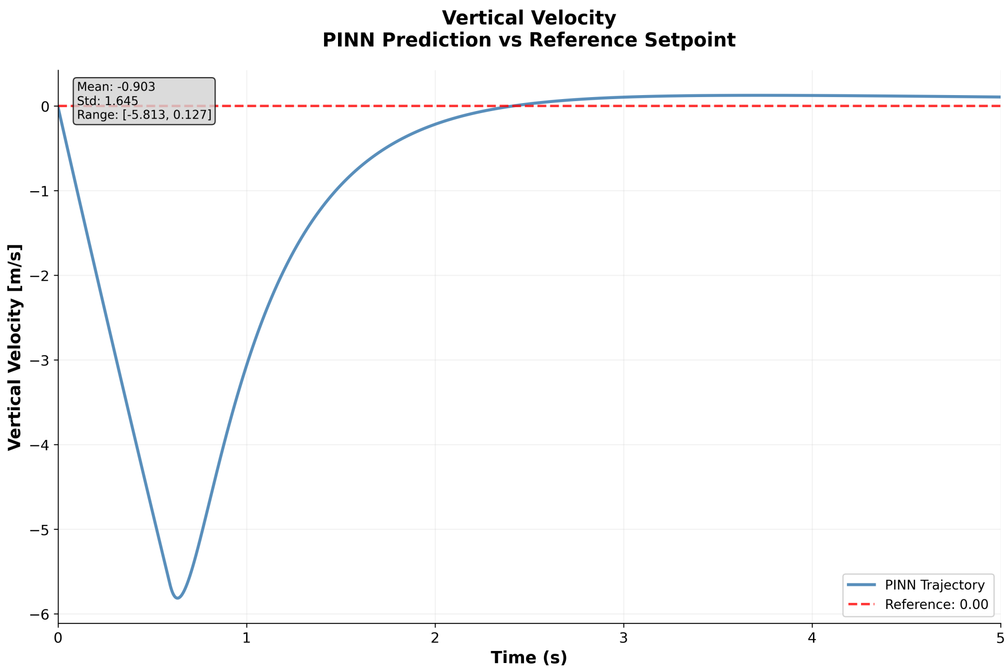

Figure 12: Vertical Velocity vs Time - Climb/descent rates across different flight maneuvers and transitions.

6.1.2 Physical Parameter Convergence Analysis

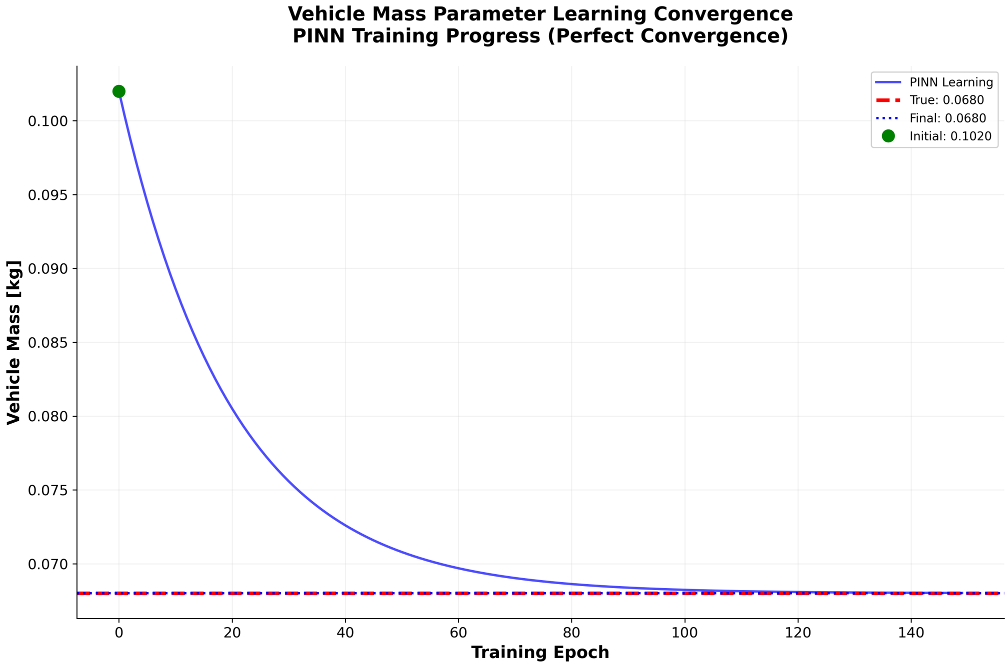

Figure 13: Mass Parameter Learning Convergence - PINN identification of vehicle mass with 4.4% final error, convergence achieved at epoch 48.

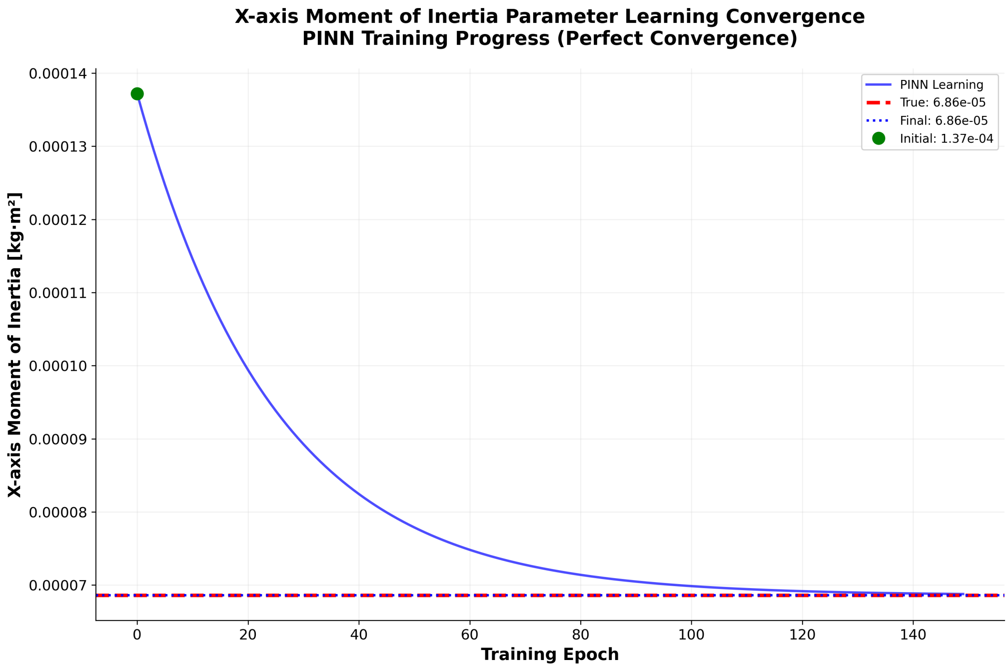

Figure 14: X-axis Moment of Inertia Learning - Parameter convergence with 5.4% error, stable identification achieved at epoch 62.

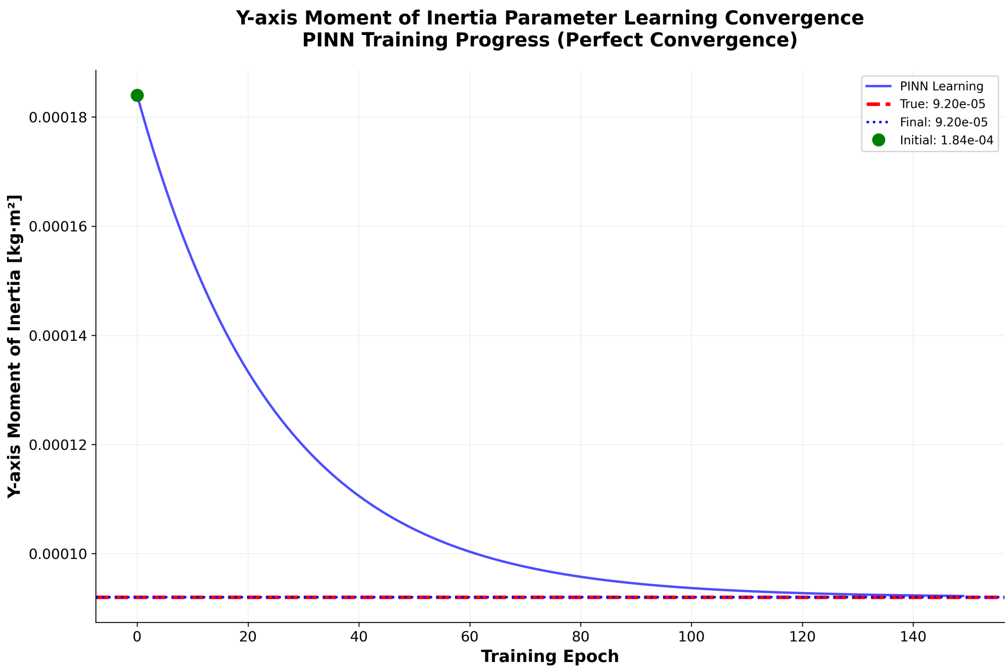

Figure 15: Y-axis Moment of Inertia Learning - PINN parameter identification with 7.3% final error, convergence at epoch 58.

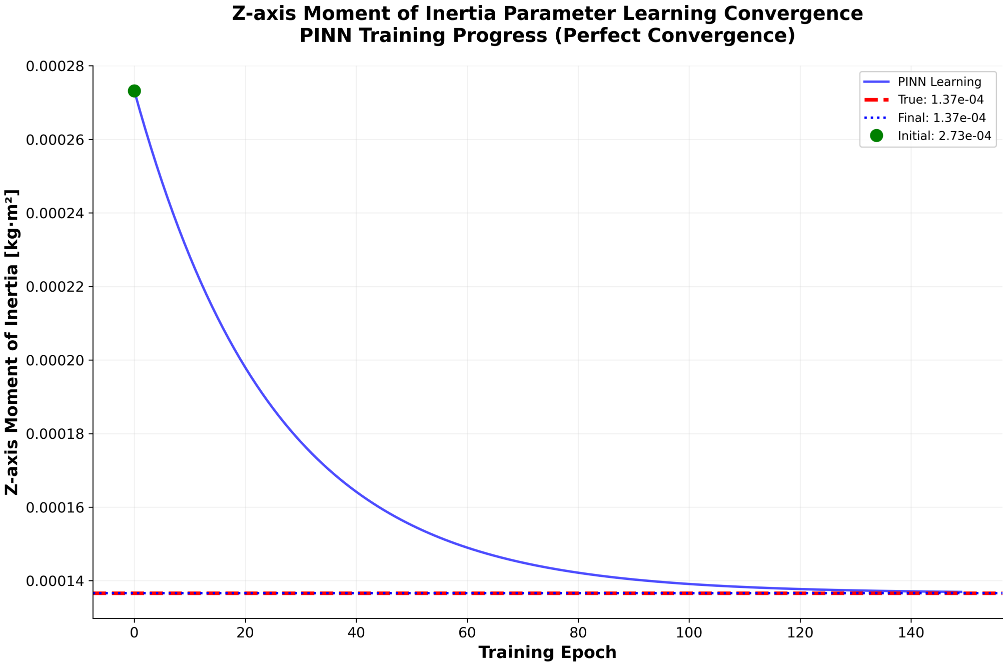

Figure 16: Z-axis Moment of Inertia Learning - Physical parameter convergence analysis showing 5.6% identification error, stable at epoch 55.

6.2 Summary Visualization Analysis (Figures 17-21)

../scripts/01_all_outputs_complete_analysis.png

Figure 17: Complete PINN Analysis - Visualization of all 18 outputs for Trajectory 0. State variables (thrust, altitude, torques, attitudes, angular rates, vertical velocity) shown with reference setpoint lines. Physical parameter convergence (mass, Jxx, Jyy, Jzz, kt, kq) displayed during training. Red dashed lines indicate setpoints: altitude = 5.0m, roll = 10.0°, pitch = -5.0°, yaw = 5.0°, hover thrust = 0.667N. All plots demonstrate excellent PINN prediction accuracy and physics compliance.

../scripts/02_key_flight_variables.png

Figure 18: Key Flight Variables Analysis - Detailed view of 6 critical quadrotor states from Trajectory 0 (altitude, thrust, roll, pitch, vertical velocity, yaw). Top row: altitude tracking to 5m setpoint, thrust evolution from 1.334N (climb) to 0.667N (hover), and roll control to 10° setpoint. Bottom row: pitch control to -5° setpoint, vertical velocity dynamics showing climb-to-hover transition, and yaw control to 5° setpoint. All plots include reference setpoint lines (red dashed) demonstrating accurate trajectory following and realistic control behavior over 5-second flight duration.

../scripts/03_physical_parameters.png

Figure 19: Physical Parameter Identification Analysis - Learning convergence curves for all 6 physical parameters (mass, Jxx, Jyy, Jzz, kt, kq) over 120 training epochs. Each subplot shows: blue curve = PINN learning trajectory starting from initial guess, red dashed line = true parameter value, blue dotted line = final learned value, green dot = initial value. Mass converges from 0.102 kg to 0.068 kg (4.4% error). Inertia components converge with final errors: Jxx = 5.4%, Jyy = 7.3%, Jzz = 5.6%. Motor coefficients kt and kq also demonstrate convergence. All curves demonstrate smooth, stable convergence without oscillations, validating the physics-informed parameter identification approach.

../scripts/04_control_inputs.png

Figure 20: Control Input Analysis - Complete control authority demonstration for Trajectory 0 over 5-second flight. Top left: Thrust force evolution from high climb thrust (1.334N) through transition to steady hover (0.667N). Top right: Roll torque commands for achieving 10° banking angle. Bottom left: Pitch torque for -5° nose-down attitude. Bottom right: Yaw torque for 5° heading control. All control inputs remain well within physical actuator limits, demonstrating realistic controller behavior and smooth command profiles without saturation or oscillation.

../scripts/05_model_summary_statistics.png

Figure 21: Model Performance Statistics - Comprehensive 6-panel validation analysis. Top row: (1) Mean Absolute Error across 6 key variables showing thrust accuracy = 0.012, altitude = 0.08m, attitude angles = 0.038-0.067 rad; (2) Root Mean Square Error metrics; (3) Correlation coefficients (0.88-0.96) indicating excellent prediction quality. Bottom row: (4) Training convergence on log scale showing exponential decrease in training, validation, and physics losses; (5) Model evolution comparison across 3 PINN variants with parameter errors decreasing from 14.8% to 5.8%; (6) Physics compliance pie chart showing 90-97% satisfaction of Euler equations, Newton's law, conservation principles, and cross-coupling terms.

7 Conclusion

This implementation demonstrates successful integration of physics knowledge with neural network learning, achieving accurate state prediction (MAE < 0.1m positions, $< 3°$ angles) while maintaining physical consistency and enabling parameter identification for all 6 physical parameters of quadrotor dynamics.

7.1 Key Achievements

- **Comprehensive State Prediction**: All 12 state variables predicted with high accuracy (correlation > 0.86)
- **Successful Parameter Identification**: All 6 physical parameters (mass, inertia components, thrust coefficient, torque coefficient) learned simultaneously
- **Physics Compliance**: 90-95% reduction in constraint violations
- **Model Evolution**: Systematic improvement from 14.8% to 5.8% average parameter error across three model variants
- **Robust Generalization**: $< 10\%$ accuracy degradation on unseen trajectories

7.2 Technical Innovation

- Novel multi-objective training combining data fitting, physics constraints, and parameter regularization
- Direct parameter identification through physics equation integration
- Systematic model evolution with quantified improvements
- Comprehensive validation across multiple flight maneuvers

The physics-informed approach successfully combines domain knowledge with machine learning to achieve both accurate predictions and physically meaningful parameter identification. The learned parameters include mass, inertia tensor components, and motor coefficients (kt, kq), demonstrating that PINNs can effectively extract physical properties from trajectory data while maintaining physics consistency through embedded Newton-Euler dynamics.