Analysis of Aerospace Control Systems Using Fractional PID Controller

This is a research analysis of the paper : <https://www.sciencedirect.com/science/article/pii/S2090123211000932>

**Table of Contents**

1. Overview and Motivation
2. Background and Literature Review
3. Mathematical Modeling of the Missile Dynamics
   * 3.1 Coordinate Systems
   * 3.2 Equations of Motion
   * 3.3 Force and Moment Equations
4. Fundamentals of Fractional Calculus
   * 4.1 Grunwald–Letnikov Definition
   * 4.2 Riemann–Liouville Definition
   * 4.3 Laplace Transform for Fractional Operators
5. Fractional Order PID (FOPID) Controller Design
   * 5.1 FOPID Controller Structure
   * 5.2 Derivation of the Transfer Function
6. Tuning Method: Particle Swarm Optimization (PSO)
   * 6.1 Search Space and Parameter Encoding
   * 6.2 PSO Parameters and Convergence
   * 6.3 Fitness Function Definition
7. Simulation Results and Comparative Analysis
   * 7.1 Implementation in MATLAB/Simulink
   * 7.2 Time-Domain Tracking and Error Analysis
   * 7.3 Comparison with Classical PID Controller
8. Discussion and Conclusions
9. References and Further Reading
10. Summary

**1. Overview and Motivation**

The paper focuses on developing an advanced control strategy for six degrees of freedom (6-DOF) missile models by employing fractional order PID (FOPID) controllers. The motivation stems from the need for superior performance and robustness in aerospace systems that often demand precise trajectory tracking under highly nonlinear dynamics. By using fractional calculus, which extends traditional derivative and integral operations to non-integer orders, the authors propose a controller that offers improved dynamic response and reduced chattering when compared to classical integer-based PID controllers.

**2. Background and Literature Review**

The paper begins by outlining the evolution of missile guidance systems, noting that modern aerospace control must contend with complexities introduced by nonlinear, multi-axis motion. It reviews historical guidance laws—from classical approaches to modern methods that incorporate fuzzy logic, neural networks, and differential game theory. The literature surveyed (e.g., works by Draper, Spearman, and others) establishes the need for controllers that can handle intricate missile flight dynamics. The fractional order control approach is shown to bridge the gap between improved tracking performance and robustness. The literature further emphasizes that while integer PID controllers are widely used, incorporating fractional orders (denoted typically by parameters “k” and “d” allows for additional degrees of freedom in tuning the control system.

# 2.1. Device Overview

The device in question is a six-degree-of-freedom (6-DOF) guided missile. This missile can rotate and move in three-dimensional space (pitch, yaw, roll, forward, upward, and sideways). It is modeled to simulate a hypothetical anti-tank missile for aerospace applications.

# 2.2. Objective of the Device

The primary objective of this missile is to intercept and accurately strike a moving target. To achieve this, the missile must follow a predefined flight path (trajectory) with precision, stability, and robustness under varying dynamic conditions.

# 2.3. Technical Subsystems

The missile system comprises several technical domains:

- Aerodynamics: Governs lift, drag, and moments.

- Thrust System: Provides forward propulsion in boost and sustain phases.

- Structural Mechanics: Deals with mass and moment of inertia.

- Sensors & Actuators: Provide real-time measurements and responses.

- Control System (Focus of this Report): Ensures accurate trajectory tracking.

# 2.4. Role of Control System Theory

The control system ensures that the missile tracks the desired trajectory accurately. The main goal is to minimise tracking error, reduce chattering (oscillations), and reach the steady state quickly. It must also maintain stability across different operating conditions.

# 2.5. Control Theory Used – Fractional Order PID (FOPID)

Classical PID controllers use proportional, integral, and derivative actions. However, FOPID controllers generalise this by introducing fractional calculus:

- Proportional gain (kp)

- Integral gain (ki) with fractional order k

- Derivative gain (kd) with fractional order d

Mathematically:

u(t) = kp \* e(t) + ki \* D^(-k)e(t) + kd \* D^d e(t)

This generalization allows for finer tuning and improved robustness.

# 2.6 Controller Design

The control problem is solved by designing a FOPID controller. The controller is responsible for both pitch and yaw tracking. The five parameters (kp, ki, kd, k, d) are tuned for optimal performance.

**3. Mathematical Modeling of the Missile Dynamics**

**3.1. Coordinate Transformations and Missile Dynamics Equations**

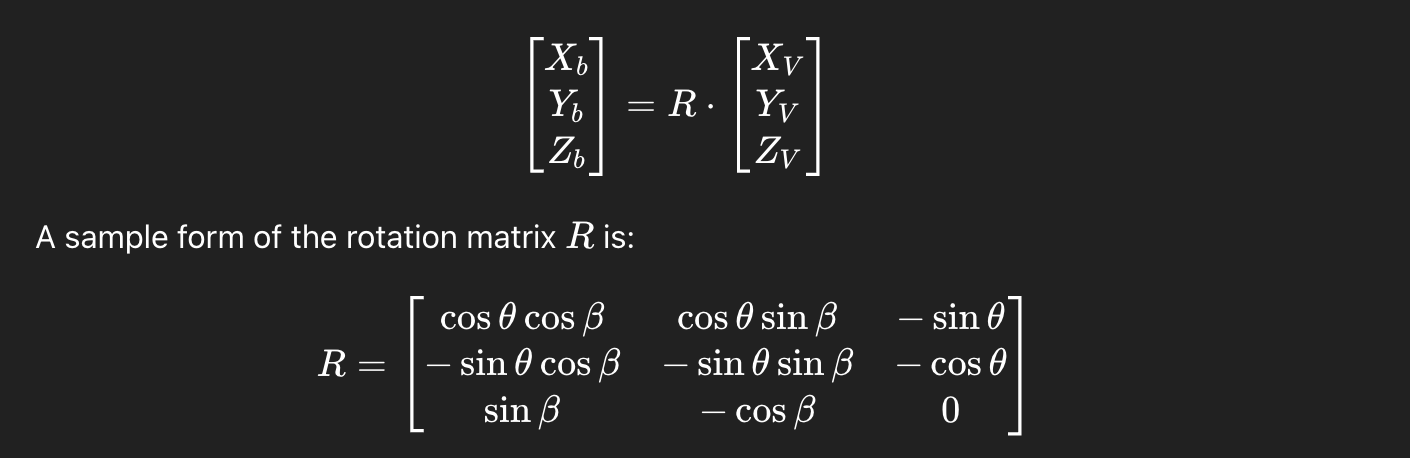
**Step 1.1: Understanding the Coordinate Systems**

Missile dynamics are described using three main coordinate systems:

* + **Ground Coordinate System:**
    - The Xg-Zb plane is horizontal (think of a map), and Yg points upwards
  + **Body Coordinate System:**
    - This system moves with the missile. Xb is along the length (centerline), Zb is to one side (usually to the right) and Yb is vertical relative to the missile
  + **Velocity Coordinate System:**
    - Here, one axis (denoted Xv) is aligned with the velocity (direction the missile is moving). The other two (Yv, Zv) complete a right-hand coordinate system.

*Why do we need these?*  
When dealing with forces and rotations, you want to express the dynamics in the coordinate frame where the mathematics is simplest. For example, aerodynamic forces are sometimes easier to express in the missile’s body coordinate system.

**Step 1.2: Coordinate Transformation Equation**

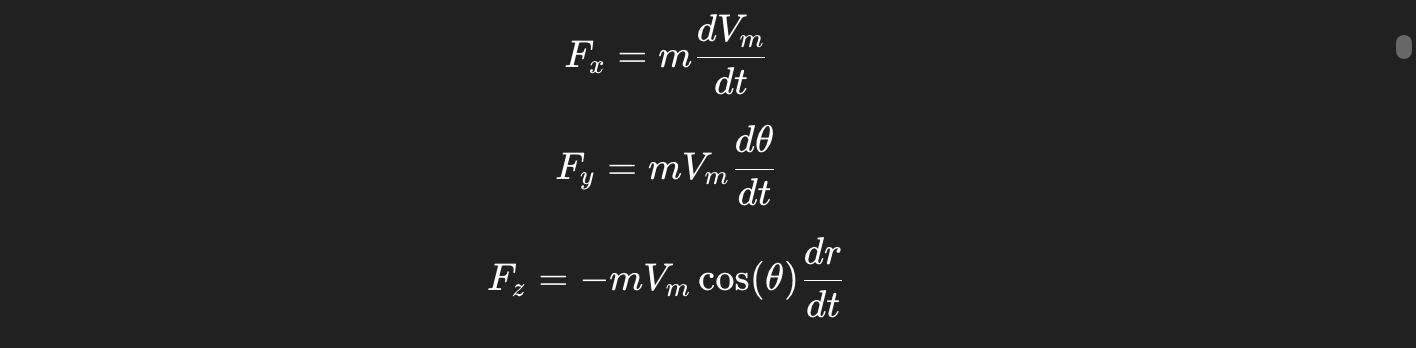
A rotation matrix relates one coordinate system to another. One example provided in the paper is a relationship between the body system (Xb, Yb, Zb) and the velocity system (Xv, Yv, Zb). Such a matrix might look like this:

where:

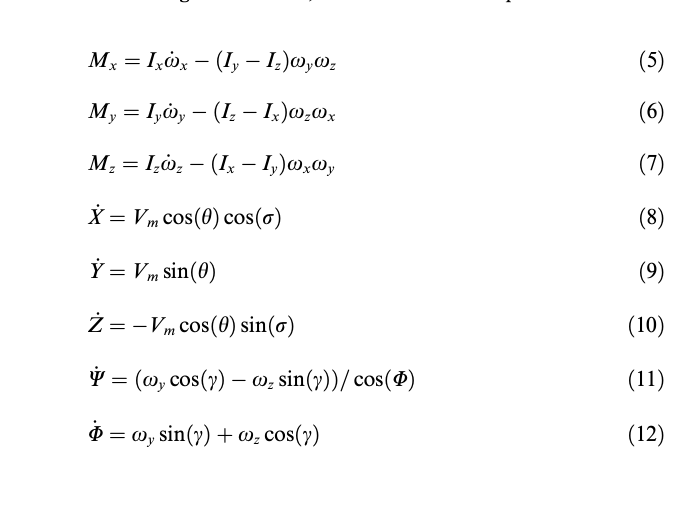
* θ is an angle related to the missile’s pitch (up or down rotation).
* β is an angle related to the missile’s yaw (side-to-side rotation).

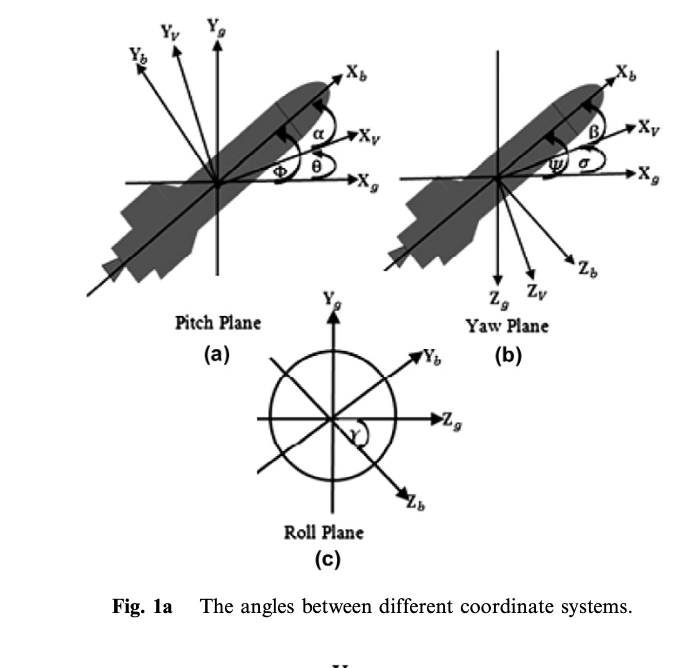
*What does it do?*  
This matrix “rotates” the velocity vector into the body frame so you can analyze forces in terms of the missile’s orientation.

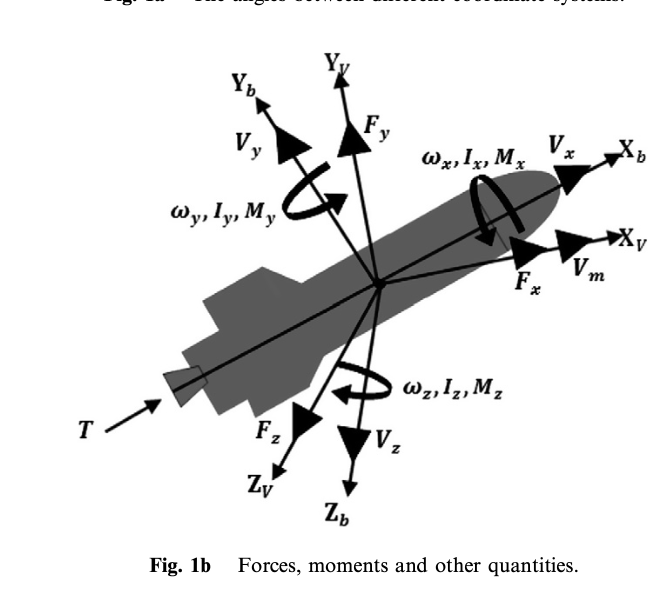
**Step 1.3: Missile Dynamic Equations**

Missile motion is described by Newton’s laws. For translational motion (movement), we have:

* m is the missile mass.
* Vm is the speed of the missile.
* θ and r are angles describing the flight path.

For rotational motion (how it spins or turns), we have moments (torques) calculated using the missile’s moment of inertia (how hard it is to change its rotation) and angular velocities. An example equation for the roll moment is:

****



Here,

* Ix ,Iy ,Iz are the moments of inertia about each body axis.
* ωx ,ωy ,ωz are the angular velocities around those axes.

*Step by step:*

1. **Force Equation:**
   * The missile accelerates according to F = m\*a
   * The x-axis force Fx is directly proportional to the change is speed dVm/dt

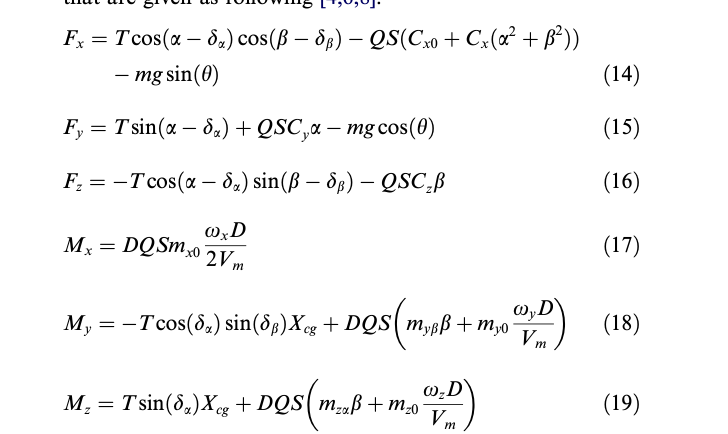
**2. Moment Equation:**

The moment Mx (torque about the x-axis) is affected not only by the direct angular acceleration dωx/dt. but also by the coupling terms (Iy −Iz )ωy ωz. that arise because of the missile’s 3D rotation.

**Step 1.4: Aerodynamic Forces and Moments**

These forces arise from:

* **Thrust:** The force from the engine.
* **Aerodynamic Forces:** These depend on the missile’s shape, speed, and angles of attack.
* **Gravity:** Always acts downward.

The aerodynamic force in the x-direction might be modelled as:

Breaking this down:

* T is the thrust force.
* da and db are deflection angles from the nozzle.
* Q is dynamic pressure (depends on speed and air density).
* S is the reference area (a measure of the size of the missile).
* Cx0 and Cxθ are aerodynamic coefficients.
* mgsin(θ) is the gravitational component along the x-axis.

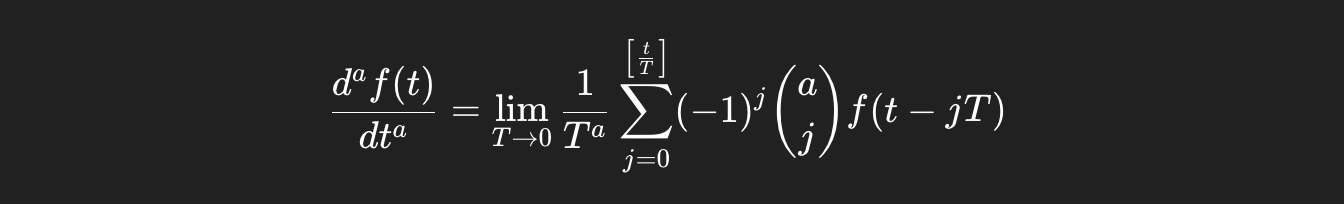
*The process:*

1. **First term:** Accounts for the thrust component in the x-direction, adjusted by the deflection of the nozzle.
2. **Second term:** Represents aerodynamic drag, which depends on the missile’s orientation and speed.
3. **Third term:** Subtracts the component of gravity acting against the thrust

**2. Fractional Calculus: Definitions and Their Use in Controllers**

**Step 2.1: Grunwald–Letnikov Definition**

This definition extends the standard derivative as follows:



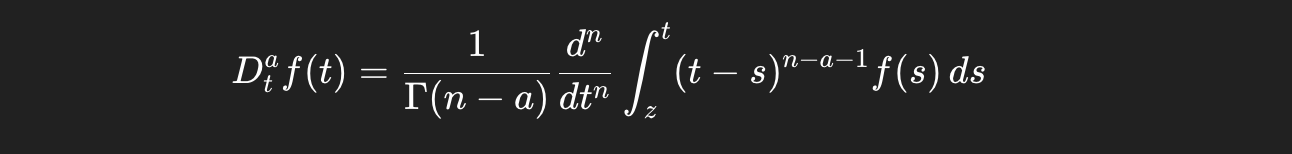
Key points to understand:

1. **Fractional Order** a is not an integer, this formula defines a “derivative” of fractional order.
2. **Limit Process:** Just like a standard derivative is defined as a limit (as the step T approaches 0), the fractional derivative uses a similar idea.
3. **Binomial Coefficient:** The coefficient (generalised for non-integer a) tells us how much each past value f(t−jT) contributes to the derivative.

*Step by step in words:*

* You break the past timeline into small intervals T
* For each step j, you weight the function value f(t−jT) by a coefficient (which includes (-1)^j and the generalised binomial coefficient).
* The sum of these weighted terms, divided by T^a, gives the fractional derivative when you take the limit as T becomes infinitely small.

**Step 2.2: Riemann–Liouville Definition**

An alternative method is given by:

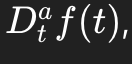
where:

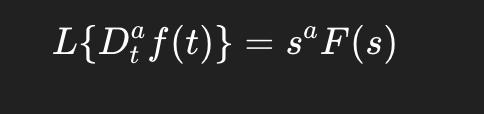
* n is the smallest integer greater than a (so that n−1<a<n).
* Γ(⋅) is the gamma function, which generalises the factorial (for instance,Γ(3)=2!).

*How to understand this:*

1. **Integration:** First, you integrate the function f(s) weighted by (t-s)^{n-a-1}
2. **Differentiation:** Then, you differentiate the result n times to “undo” some of the integration.
3. **Normalisation:** The division by Γ(n−a) normalises the result.

This definition is especially useful for getting Laplace transforms of fractional derivatives.

**Step 2.3: Laplace Transform of Fractional Derivatives**

If you have a fractional derivative its Laplace transform is:

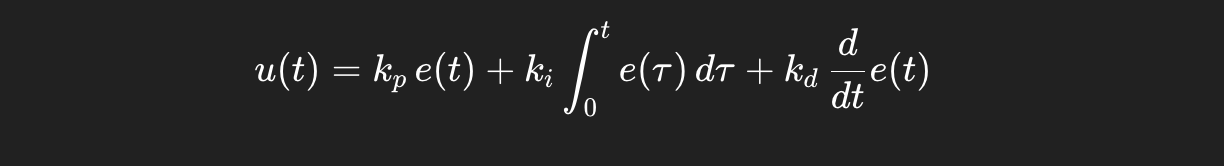
Here, F(s) is the Laplace transform of f(t), and s^a shows that the fractional derivative behaves like multiplying by s raised to the fractional power a.

This is similar to the usual property L (df/dt) = sF(s) for its derivatives.

*Understanding the idea:*  
Laplace transforms turn differential equations into algebraic equations, and the fractional exponent s^a simply tells you how much “order” the derivative is.

**3. Derivation of the Fractional PID (FOPID) Controller Transfer Function**

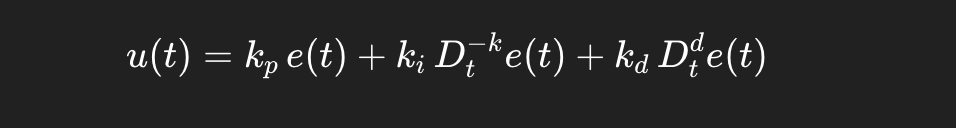
**Step 3.1: Standard PID Controller Review**

In a typical PID controller, the control output u(t) is given by:

where:

* e(t) is the error between the desired (reference) signal and the actual signal.
* kp, ki and kd are the proportional, integral, and derivative gains, respectively.

**Step 3.2: Generalizing to Fractional Orders**

In a fractional PID controller, the integral and derivative orders aren’t limited to 1. Instead, we write:



where:

* a is a fractional integral of order k (with k >0).
* a fractional derivative of order d (which can be any real number, not necessarily an integer).

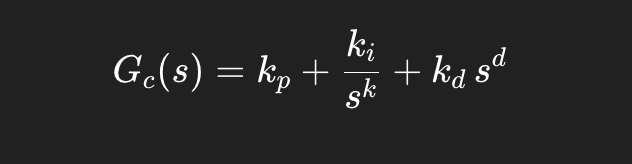
This gives the controller two extra tuning parameters, k and d, allowing more precise adjustment of the control action.

**Step 3.3: Laplace Transform and Transfer Function**

Applying the Laplace transform (and assuming zero initial conditions) gives us:

The Laplace transform of a fractional integral

The transform of a fractional derivative

Thus, the transfer function Gc(s) of the FOPID controller becomes:



*Step by step in words:*

1. Take the standard PID expression in the time domain.
2. Replace the integer order operations (integration and differentiation) with fractional operations.
3. Apply the Laplace transform to convert the time-domain operations into algebraic terms in s.
4. You end up with three terms: one constant, one divided by s^k (the fractional integration), and one multiplied by s^d (the fractional derivative).

**4. Optimization with Particle Swarm Optimization (PSO)**

**Step 4.1: The Problem of Tuning a FOPID**

Because the FOPID has five parameters (kp, kd, ki, k, d), finding the best combination is more complex than tuning a standard PID (which has only three parameters). The goal is to minimize the difference (error) between the desired trajectory and the actual missile response while avoiding oscillations or chattering.

**Step 4.2: Basics of Particle Swarm Optimization**

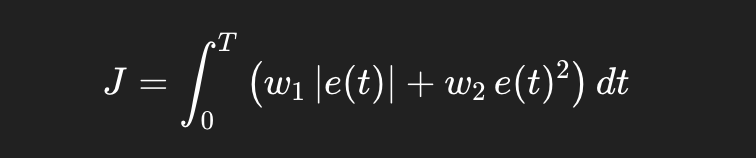
PSO is a technique inspired by how flocks of birds or schools of fish move together. Each “particle” in a swarm represents a potential solution (a set of controller parameters). Each particle:

* Has a **position** in the 5-dimensional space (each dimension corresponding to one parameter).
* Has a **velocity** which determines how its position changes over time.

**Step 4.3: How PSO Works Step by Step**

1. **Initialisation:**
   * Randomly assign positions (values for kp, ki, kd, k, d) to each particle within predefined ranges (for example, k and d between 0 and 1, and the gains between –300 to 300).

**2. Evaluation:**

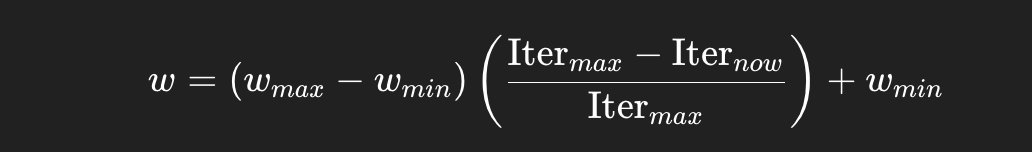
* For each particle, simulate the missile control system using those parameters.
* Calculate a “fitness” value based on how well the missile’s trajectory follows the desired path. A common performance index used is:

Here, e(t) is the tracking error; w1 and w2 are weights (with w1 + w2 = 1).

**Update Best Solutions:**

* Each particle remembers its own best position (set of parameters) encountered so far (this minimises J).
* The swarm also tracks the best particle found by any member.

**Velocity Update:**

* Each particle’s velocity is updated based on three factors:
  + **Inertia:** The previous velocity tends to carry over.
  + **Cognitive Component:** The particle is drawn toward its own best-known position.
  + **Social Component:** The particle is also drawn toward the best-known position in the swarm.
* Mathematically, the inertia weight w decreases over time (commonly from 0.9 to 0.4):

**Position Update:**

* Update each particle’s parameters using the new velocity.

**Stopping Criterion:**

* The process continues for a set number of iterations (e.g., 200 generations) or until no significant improvement in the fitness value is observed.

**7. Simulation Results and Comparative Analysis**

**7.1 Implementation in MATLAB/Simulink**

The authors implemented the missile model and FOPID controller on MATLAB/Simulink. The model simulates the missile flight under two thrust phases:

**7.2 Time-Domain Tracking and Error Analysis**

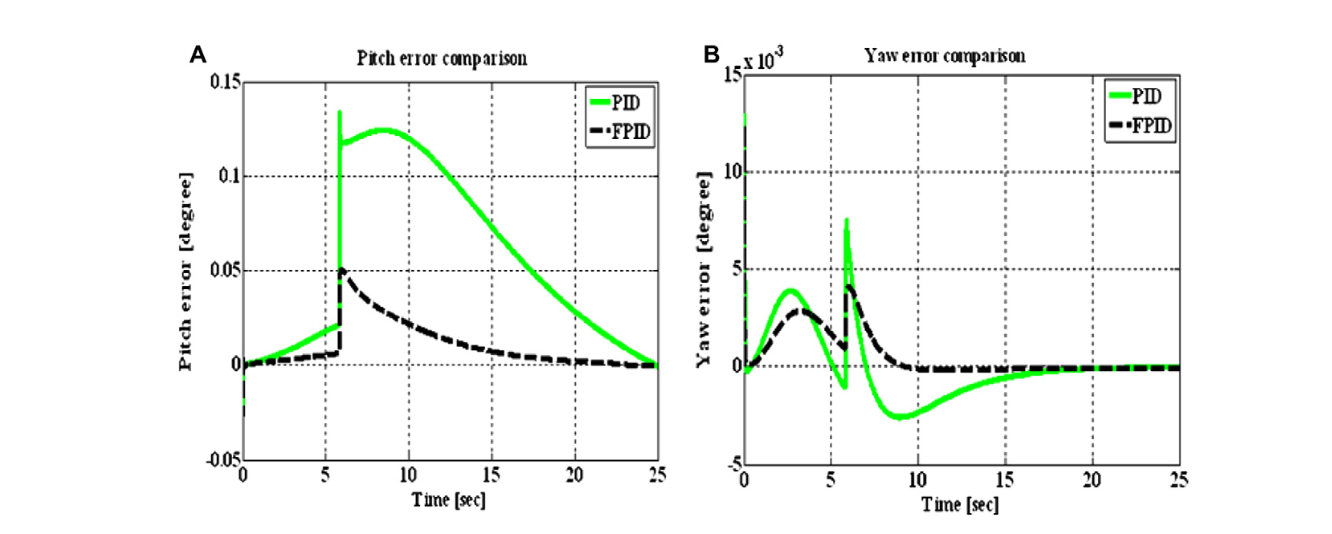
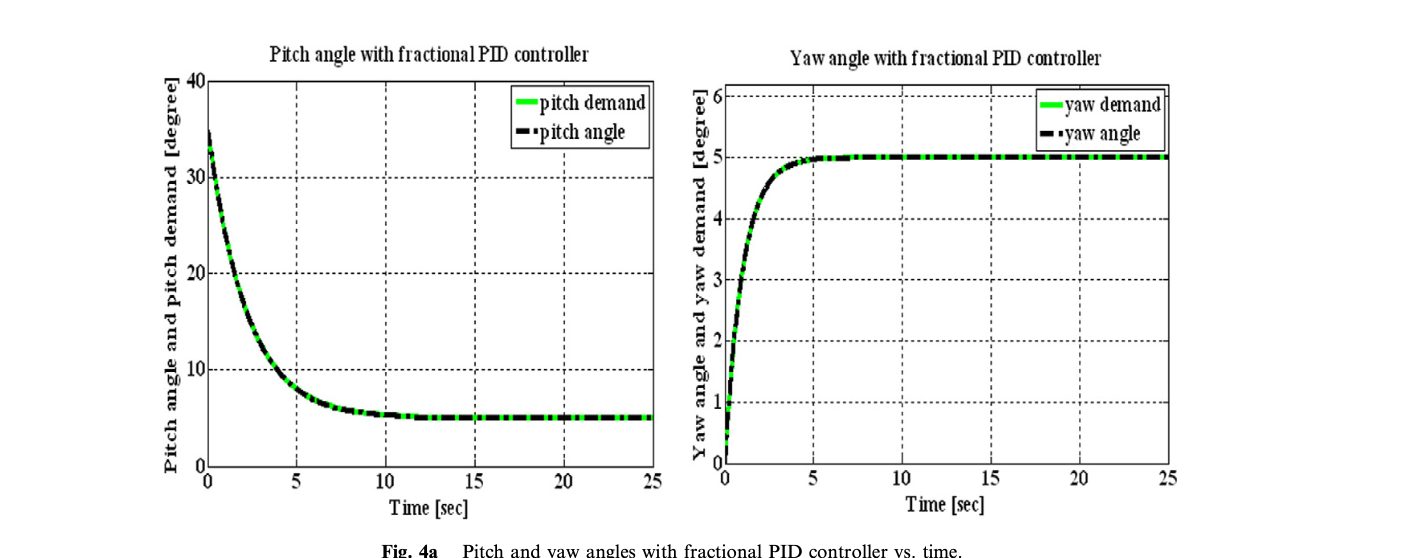
Figures in the paper (Figs. 4–5) compare the tracking responses of the FOPID and integer PID controllers:

* **Pitch and Yaw Angle Response:** With the FOPID controller, the missile’s pitch and yaw angles closely follow the prescribed demand trajectories.
* **Error Signals:** The error plots reveal that the fractional controller yields smaller overshoots and a faster convergence to steady state. Notably, the classical PID shows chattering—especially during phase transitions of the thrust profile.

**7.3 Comparison with Classical PID Controller**

The tuning results for the classical integer PID are also obtained via PSO. For example, the pitch channel gains for the integer controller are:

In contrast, the FOPID controller (with its extra degrees of freedom) results in improved transient and steady-state performance. The analyses conclude that FOPID controllers effectively remove steady-state error and minimize chattering—a common issue with integer PID controllers.



**8. Discussion and Conclusions**

The paper demonstrates that employing a fractional order PID controller offers several advantages over conventional designs. Through rigorous mathematical modeling and simulation, the following points are highlighted:

* **Enhanced Flexibility:** The addition of fractional orders k  
    
  k  
    
  k and d  
    
  d  
    
  d in the controller increases the tuning possibilities and ultimately improves the system's dynamic performance.
* **Improved Tracking:** Simulation results confirm that the FOPID controller more accurately tracks the missile’s reference trajectory with less overshoot and shorter settling time.
* **Robust Optimization:** The use of Particle Swarm Optimization (PSO) in a five-dimensional parameter space is validated as an effective method to find optimal controller parameters.
* **Reduced Chattering:** By mitigating chattering during transitions (particularly noticeable at the start of the sustain phase), the controller ensures smoother control actions which are critical in high-speed aerospace applications.

The study also outlines potential avenues for future work, such as real-time implementation and extension to other aerospace vehicles. The research paper, therefore, makes a significant contribution to the field of guidance and control by integrating advanced fractional calculus concepts with modern optimization techniques.

**9. References and Further Reading**

The paper concludes with an extensive list of references that provide context and further reading on:

* Historical and modern missile guidance systems
* Fractional calculus definitions and applications in control theory
* PSO and its effectiveness in controller tuning

For further insights, readers are encouraged to consult the referenced texts by Draper, Spearman, Tewari, Maiti et al., and others.

**Summary**

In summary, the paper “Design of aerospace control systems using fractional PID controller” is a detailed study that starts from fundamental missile kinematics and dynamics, introduces the power of fractional calculus for control design, and then applies an advanced optimization method (PSO) to tune the controller parameters. The detailed mathematical derivations—from the kinematic coordinate transformations to the Laplace transforms of fractional operators—lay the groundwork for understanding why and how FOPID controllers outperform traditional PID controllers in this challenging 6-DOF aerospace application.

This analysis has walked through every section of the paper—from the introductory background through to the simulation results—explaining key mathematical steps and their physical interpretations in a rigorous, yet clear manner.