International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 15, Issue 11 [November 2026] PP: 264-269

Applying Artificial Intelligence-Assisted Problem-Based Learning In Teaching Mechanics To Information Technology Students

ThS. HUỲNH THỊ THANH TUYỀN
Trường Đại học Công nghệ thông tin và truyền thông Việt – Hàn, Đại học Đà Nẵng

M.A. HUỲNH THỊ THANH TUYỀN[1]
Vietnam – Korea University of Information and Communication Technology – Da Nang university

Abstract
This paper introduces a significant pedagogical approach for enhancing learner engagement: utilizing Artificial Intelligence (AI)-assisted Problem-Based Learning (PBL) to connect lesson content with interdisciplinary real-world applications, thereby fostering students' interest in Physics. Through this study, the author designs a Physics teaching process integrating PBL with AI in the Mechanics module, where AI serves as a tool for simulation, hypothesis verification, and conceptual critique. Accordingly, the paper proposes a four-step procedure, AI governance mechanisms, and academic integrity guidelines to develop physical thinking competencies among Information Technology (IT) students at the Vietnam – Korea University of Information and Communication Technology (VKU) within the current educational context.
Keywords*: Problem-Based Learning (PBL); Artificial Intelligence (AI); Physics; Information Technology students.*

Date of Submission: 12-11-2026 Date of acceptance: 25-11-2026

I. Introduction

In response to the increasingly high demands for human resource quality in the Industry 4.0 era and Vietnam's international integration, continuous learning is essential—requiring individuals to maximize internal strengths, apply scientific knowledge to daily life, and break free from rigid, stereotypical thinking and actions. Therefore, active learning and creative thinking must be cultivated and nurtured at all times. In undergraduate IT programs, students generally possess strong algorithmic thinking and high enthusiasm for technology, yet they occasionally undervalue General Physics courses due to a perceived lack of direct applicability to programming. As an experimental science, General Physics provides the foundation for systems thinking, mathematical modeling, and understanding the operating principles of hardware, sensors, signal transmission, and physical computing. However, teaching practice reveals that students easily fall into mechanical formula memorization when problems lack an interdisciplinary, real-world context.

Simultaneously, the rapid advancement of AI and intelligent computing tools opens up opportunities for pedagogical innovation. Research on Physics education and AI indicates that AI tools can support concept explanation, suggest problem-solving approaches, and simulate scenarios. Nevertheless, they also impose strict requirements regarding accuracy verification, image or chart processing, and preventing over-reliance that could diminish students' independent thinking. Consequently, integrating AI into PBL must be structured systematically rather than applied spontaneously. Employing AI-assisted PBL to link course content with interdisciplinary practice is an optimal measure to stimulate student interest. Drawing upon personal teaching experience, this paper investigates AI-assisted PBL in Physics and applies it to design a teaching process for the Mechanics section of General Physics at the Vietnam – Korea University of Information and Communication Technology.

Email: htttuyen@vku.udn.vn

II. Theoretical Basis of Problem-Based Learning and Characteristics of IT Students

*** Problem-Based Learning (PBL) in Physics** [2]: PBL confronts students with a practical situation or an open-ended problem without a pre-existing solution. Learners must independently identify the problem, mobilize background knowledge, formulate hypotheses, select physical models, execute solutions, and evaluate outcomes

***Characteristics of IT Students Learning Physics**: Alongside their advantages—familiarity with analytical problem-solving and working with variables, algorithms, charts, and digital tools—IT students often exhibit limitations such as overlooking boundary conditions, measurement units, or the physical meaning of parameters when focusing solely on algebraic manipulation or code execution. Their learning motivation increases significantly when Physics problems are tied to technology-relevant topics, such as kinematics in game development and graphics, oscillations and waves in telecommunications, or electromagnetism in microchips and sensors.

*** Roles of AI and Lecturers in the PBL Process** [1]: Within this model, AI acts as a catalyst for discovery and critical review rather than a substitute for student cognition—suggesting problem scenarios, assisting in writing simulation scripts (e.g., Python), or cross-checking arguments. Meanwhile, the lecturer takes on the role of designing problems, guiding methodology, ensuring scientific accuracy, and training students in information evaluation.

Essential principles to follow include defining learning objectives, content, and an interdisciplinary knowledge system alongside supporting learning materials. Furthermore, during instruction, lecturers must ensure that applications are feasible, contextually appropriate, not overused, engaging, selective, scientifically accurate, aligned with student proficiency, timely or widely applicable, and conducive to a comfortable, engaging learning environment.

III. Applying AI-Assisted Problem-Based Learning in Teaching General Physic

General Physic belongs to the Department of Natural Sciences and constitutes part of the foundational knowledge block in undergraduate curricula. Based on curriculum analysis, the author effectively implemented AI-assisted PBL across several lessons. Utilizing the proposed measures, illustrative lesson plans were designed for topics including projectile motion and collisions, rotational dynamics of rigid bodies and moment of inertia, and conservation of angular momentum and gyroscopes within the Mechanics module. These designs aim to stimulate interest and enhance creativity among students at VKU. The implementation details are as follows:

- Implementation Steps for AI-Assisted PBL

The teaching process is structured into four distinct steps:

1. Problem Identification: The lecturer presents an interdisciplinary technical scenario; students analyze the given data and define the core physical problem.

2. Model Construction with AI: Students establish physical equations or hypotheses and use AI to verify applicability conditions or write numerical simulation code.

3. Execution, Cross-Checking, and Critique: Students solve the problem theoretically combined with simulation, cross-check AI-generated outputs, and identify logical or physical flaws.

4. Reporting and Knowledge Standardization: Student groups present their solutions alongside their AI interaction logs; the lecturer consolidates core knowledge and conducts assessments.

- Lesson Plan Design in "Rigid Body Dynamics" (Mechanics)

Two lesson plans were selected and developed for this chapter, accompanied by a comparative analysis demonstrating the superiority of PBL over traditional lesson plans.

*** Content Structure of "Rigid Body Dynamics"**: This is a crucial topic in Mechanics, encompassing composition and resolution of forces, equilibrium conditions of rigid bodies, projectile motion, collisions, equations of rotational motion, angular momentum, torque, moment of inertia, conservation of angular momentum, and gyroscopes. This knowledge is applied to solve rigid body mechanics problems and explain scientific and technological phenomena. Studying this content builds both academic foundations and practical competence applicable to today's rapidly evolving IT sector.

*** Identifying Chapter Problems for AI-Assisted PBL**: Based on learning objectives, content, and student proficiency, the author proposes the following core topics:

- Projectile motion and collisions in Game Programming or 3D Graphics.
- Rotational dynamics of rigid bodies and moment of inertia (applied in Robotic Arm design and LiDAR systems).
- Conservation of angular momentum and gyroscopes (applied in IMU sensors on Smartphones/Drones).

*** Development of Two Illustrative AI-Assisted PBL Lesson Plans**

Lesson 1: "Projectile Motion and Collisions in Game Programming or 3D Graphics" (2 Periods)

- *Core Content*: Establishing trajectory equations under gravity and air resistance; applying conservation of momentum and the coefficient of restitution in collisions. Using AI to convert analytical models into time-stepped (Δ t) discrete models in Game Physics Engines and detecting cumulative errors.

- *Problem Scenario* (IT Context):

Scenario: When programming cannonball trajectories or object throwing in a 3D game on terrain with headwinds, developers often use the simple Euler method $v_{t+\Delta t} = v_t + a\Delta t$. However, when the frame rate drops from 60 FPS to 20 FPS, the projectile misses its target and passes straight through thin walls (tunneling).

Task: Design a physical model for an object of mass m = 2 kg launched at an angle α = 45° with an initial velocity v_0 = 30 m/s in a medium with air resistance $\overrightarrow{F_c} = -k\vec{v}$ (k = 0.1 kg/s), and propose a solution to prevent tunneling during a partially elastic collision (coefficient of restitution e = 0.6).

- *Teaching Procedure*:

+ **Activity 1**: Problem Identification (Scenario Presentation, Grouping, and Modeling Orientation)

Lecturer's Role: Organizes students into groups and plays two short video clips comparing physics bugs in a 3D game: trajectory deviation when dropping from 60 FPS to 20 FPS, and a bullet tunneling through a thin wall. Poses a thought-provoking question: "Why does frame rate alter the physical laws governing an object despite using the exact same programming formula?" Guides students in decomposing a complex problem into basic physical phases using guiding questions about acting forces and state variables. Observes and assists groups struggling to translate programming terms into physical quantities.

Student Activities: Visually observe the physics anomalies (incorrect range, incorrect maximum height, missed collisions). Discuss and divide the problem into two independent phases: flight in air with drag and collision interaction with a barrier. Draw free-body diagrams $\vec{P}$ and $\vec{P}$, select an appropriate coordinate system (Oxy), and list initial state variables (x, y, v_x, v_y).

AI Requirements and Outputs: Intermediate output: A two-phase problem breakdown sketch on the worksheet. Final output: A complete free-body diagram and a full table of state variables (x, y, v_x, v_y) and system parameters (m, g, k, e, Δt).

+ **Activity 2**: Model and Hypothesis Construction (Establishing Equations, Finding Exact Solutions, and Model Review)

Lecturer's Role: Guides students in translating the free-body diagram into Newton's Second Law equations along the Ox and Oy axes. Asks: "Do the projections of drag force $\overrightarrow{F_c} = -k\vec{v}$ onto each axis depend independently on v_x, v_y?" Reviews the coefficient of restitution $e = -\frac{v'_n}{v_n}$. Allows groups to use AI tools to verify integral calculations for exact analytical solutions (x(t), y(t)). Enforces prompting rules: prompts must specify the reference frame, SI units, and initial conditions (x_0, y_0, v_{0x}, v_{0y}). Suggests dimensional analysis to verify analytical terms.

Student Activities: Write differential equations of motion: along Ox: $m\frac{dv_x}{dt} = -kv_x$; along Oy: $m\frac{dv_y}{dt} = -mg - kv_y$. Set boundary conditions for ground impact (y = 0) or vertical wall impact x = x_{wall} to update rebound velocities $v'_y = -ev_y$ or $v'_x = -ev_x$. Prompt AI to solve the differential equations with m = 2 kg, v_0 = 30 m/s, α = 45°, k = 0.1 kg/s, and self-verify AI results by taking the limit as k → 0 (checking if it reduces to ideal projectile motion). Record the first interaction in the AI Log.

AI Requirements and Outputs: AI is prohibited during manual equation formulation to ensure conceptual mastery. AI Role: Analytical math assistant and dimensional checker. Output: Exact analytical formulas (x(t), y(t)) serving as the benchmark for numerical simulation.

+ **Activity 3**: Numerical Simulation, Cross-Checking, and AI Critique (Discrete Simulation Programming, Tunneling Resolution, and AI Critique)

Lecturer's Role: Instructs students to use AI to write a Python script simulating trajectories using two numerical methods Explicit Euler and 4th-Order Runge-Kutta (RK4) at two time steps: Δ t_1 = 0.016s (60 FPS) and Δ t_2 = 0.05s (20 FPS). Asks: "Which curve tracks the analytical solution from Phase 2 most closely?" Triggers a challenge scenario: asks students to prompt AI for collision detection code against a thin wall at x = 40 m (thickness 0.1 m) at 20 FPS, guiding them to spot physical/algorithmic flaws in AI's code (AI typically uses discrete checks like "if x >= 40" after the object has already skipped past the wall at step t + Δ t).

Student Activities: Use AI to generate a Python script (numpy and matplotlib) plotting three trajectories on the same axes: Exact Analytical, Euler, and RK4. Measure errors in range L and maximum height H_{max} across both FPS settings and save the charts. Test AI's collision code, identify tunneling when displacement$\Delta x = v_x\Delta t$ exceeds wall thickness, and prompt AI to upgrade the algorithm to Continuous Collision Detection (CCD) or linear interpolation of impact time. Complete the AI Critique Log detailing AI's errors and the group's fixes.

AI Requirements and Outputs: AI serves as a coding/visualization assistant and an object of technical critique. Outputs: Trajectory error comparison charts across Δt, corrected Python source code resolving tunneling, and the AI Critique Log.

+ Activity 4: Reporting, Knowledge Standardization, and Assessment (Group Presentation, Debate, Standardization, and Evaluation)

Lecturer's Role: Invites two groups with differing approaches to present briefly (5 minutes/group) on trajectory charts, anti-tunneling solutions, and detected AI flaws; facilitates peer cross-questioning on energy and momentum conservation during collisions. Summarizes and standardizes three core concepts: velocity-dependent drag force and momentum conservation; the Fixed Timestep principle (decoupling the physics loop from FPS) in Game Engine architecture; and dimensional/boundary verification when using AI. Collects worksheets and AI logs and announces grading criteria.

Student Activities: Group representatives present live simulations and AI Critique Logs; peers critique accuracy and computational cost across numerical integration algorithms. Record standardized notes and submit the group portfolio (Worksheet + .py script + AI Critique Log) to the Learning Management System (LMS).

AI Requirements and Outputs: AI use is strictly halted; students defend results through personal understanding. Output: complete group learning portfolio for continuous assessment.

Lesson 2: "Rotational Dynamics of Rigid Bodies and Moment of Inertia (Applied in Robotic Arm Design and LiDAR Systems)" (2 Periods)

- Core Content: Establishing the fundamental equation of rigid body rotation around a fixed axis ($M = I\gamma$); calculating the moment of inertia I for non-uniform mass distributions and applying the Huygens–Steiner (parallel axis) theorem. Using AI to compute mass-distribution integrals, simulate angular response θ(t), and optimize servo motor torque.

- Problem Scenario (IT Context):

+ *Scenario*: A robotics engineering team designs a 1-DOF robotic arm (a uniform rod of length L = 0.6 m, mass m_1 = 1.2 kg) carrying a LiDAR sensor module of mass m_2 = 0.8 kg mounted at its tip. The joint's stepper motor provides a maximum torque M_{max} = 4,5 Nm. When programmed to lift the arm from horizontal to vertical within t = 0.5s, the motor skips steps and overloads.

+ *Task*: Calculate the total moment of inertia of the system about the rotational axis, determine the angular acceleration required to complete the lift in 0.5s, check whether the current motor can meet this requirement, and propose mechanical modifications or velocity control algorithms (trapezoidal velocity profile) to resolve the issue.

- Teaching Procedure

+Activity 1: Problem Identification (Scenario Presentation, Grouping, and Force/Torque Analysis)

Lecturer's Role: Shows a real-world video of a 1-DOF robotic arm with a LiDAR sensor jerking, vibrating, or skipping steps during rapid upward acceleration. Asks: "Why is the arm most prone to skipping steps right at the start of the lift from the horizontal position?" Divides the class into groups (4–5 students/group), distributes worksheets, assigns the open-ended motor verification task M_{max}= 4.5 Nm, L = 0.6 m, t = 0.5s), and defines the elevation angle θ relative to the horizontal. Uses guiding questions to help students locate centers of gravity and analyze changing lever arms from $\theta = 0^0$ to 90^0.

Student Activities: Observe step-skipping and jerky motion in the video; identify factors affecting motor lifting capacity (lever arm length, tip sensor mass, lifting speed). Summarize technical specifications into physical quantities (L, m_1, m_2, M_{max}, t) and identify the fixed pivot at the motor joint. Draw the free-body diagram showing rod weight $\overrightarrow{P_1}$ (at midpoint L/2) and LiDAR weight $\overrightarrow{P_2}$ (at tip L), formulate gravitational resisting torque $M_{cản}(\theta) = (m_1 g \frac{L}{2} + m_2 g L)cos\theta$, and conclude that peak load occurs at $\theta = 0^0$.

AI Requirements and Outputs: AI is prohibited to foster independent physical observation and analysis. Outputs: Parameter summary table, coordinate conventions, complete free-body diagram, and resisting torque expression at $\theta = 0^0$.

+Activity 2: Model and Hypothesis Construction (Establishing Moment of Inertia and Rotational Equations, Verification, and Assumption Review)

Lecturer's Role: Guides students in splitting the rigid system into two components relative to the joint axis: a uniform rod of length L and a point mass m_2 at the tip. Asks: "When lifting the arm from rest ($\omega_0 = 0$) to π/2 rad in 0.5s under constant angular acceleration γ, at what instant is maximum motor torque required?" Permits AI usage to verify numerical calculations (L = 0.6 m, m_1 = 1.2 kg, m_2 = 0.8 kg, t = 0.5 s, g = 9.8 m/s²) and prompts students to query AI about the Huygens–Steiner theorem and switching from constant acceleration to a trapezoidal velocity profile (acceleration – constant velocity – deceleration).

Student Activities: Derive total moment of inertia: $I = I_{thanh} + I_{LiDAR} = \frac{1}{3}m_1L^2 + m_2L^2$. Formulate the rotational dynamics equation: $M_{\text{động cơ}} - M_{\text{cản}}(\theta) = I\gamma$. Calculate minimum angular acceleration γ from $\theta = \frac{1}{2}\gamma t^2$.

AI Requirements and Outputs: AI is disallowed during manual formula derivation to reinforce conceptual understanding. AI Role: Calculation checker and trajectory control advisor. Output: Formula set and numerical proof demonstrating motor overload at startup.

+**Activity 3**: Numerical Simulation, Cross-Checking, and AI Critique (Torque Curve Simulation, Critique Traps, and AI Error Correction)

Lecturer's Role: Instructs students to use AI to write a Python script plotting required motor torque M(t) over $t \in [0; 0{,}5s]$ under two scenarios: (1) constant acceleration and (2) trapezoidal velocity profile. Asks groups: "Why does the M(t) curve decrease during the lift even when angular acceleration γ remains constant?" Triggers a critique trap: instructs students to prompt AI with the problem while intentionally omitting the rod's pivot location (causing AI to default to center-of-mass inertia $I = \frac{1}{12}m_1L^2$ instead of endpoint inertia $I = \frac{1}{3}m_1L^2$, or to neglect the arm's own weight), guiding students to audit AI's Python code line by line.

Student Activities: Generate Python scripts (numpy and matplotlib) to simultaneously simulate θ(t), ω(t), and M(t), plotting a horizontal threshold line at M_{max}= 4,5 Nm to visualize the step-skipping zone. Experiment with adjusting lift duration t or adding a counterweight to bring M(t) below M_{max}. Analyze AI's explanation and code, pinpoint physical errors (wrong pivot axis, missing $m_1g\frac{L}{2}cos\theta$), or forgetting degree-to-radian conversion in np.cos()), debug the script, and complete the AI Critique Log.

AI Requirements and Outputs: AI acts as a simulation/visualization assistant and an object of technical verification. Outputs: Comparison plots of M(t) across trajectory profiles, debugged Python code, and an AI Critique Log identifying at least one AI error.

+**Activity 4**: Reporting, Knowledge Standardization, and Assessment (Solution Debate, Standardization, and Evaluation)

Lecturer's Role: Invites two groups with distinct optimization strategies (one hardware-based, such as adding a counterweight or shortening the lever arm; one software-based, using trajectory control) to present for 5 minutes; encourages peer debate on pros and cons. Consolidates three core lessons: calculating composite moments of inertia and the role of the rotational axis; linking M = Iγ to Robotics motion control algorithms; and verifying implicit assumptions and angular units (radians vs. degrees) in AI-generated physics code. Collects portfolios and announces continuous assessment grades.

Student Activities: Present simulation charts, analyze step-skipping causes, and explain how AI errors were detected and fixed. Debate the trade-offs of counterweights (reducing gravitational torque while increasing total moment of inertia I). Record standardized notes and submit the group package (Worksheet + Python Script + AI Critique Log) to the LMS

AI Requirements and Outputs: AI use ceases; students defend arguments using physics and engineering principles. Output: Complete learning portfolio evaluating problem-solving and digital competencies

*** Assessment Criteria** (Summary Rubric)

- Physical Modeling Thinking (40%): Accurately formulating equations, strictly controlling measurement units (SI system), and explaining the physical meaning of parameters.
- Solution and Simulation Quality (30%): Producing accurate simulation graphs that satisfy the boundary conditions of the engineering problem.
- AI Utilization and Critique Competency (30%): Formulating logical prompts, detecting and correcting AI errors or implicit assumptions, and transparently citing AI interactions.

*** Challenges and Solutions for Ensuring Academic Integrity**

To ensure effective instruction, lecturers must establish clear rules: students may use AI to suggest simulation code structures and verify mathematical transformations, but they must not copy explanations verbatim without physical verification. Grading directly on the "AI Critique Log" and conducting rapid in-class oral checks curb technological over-reliance while fostering scientific critical thinking. Following the lessons, students demonstrated markedly heightened engagement because the course content directly aligned with their major, generating extensive peer-to-peer and student-lecturer discussions.

IV. Evaluation of Results and Conclusion

4.1. Empirical Evaluation and Scientific Value of the Research

After defining learning objectives and developing AI-integrated problem scenarios tailored to each topic, the author designed teaching procedures and implemented them in an IT course for Cohort 25 (K25) at the university. Assessment exams covering the taught content were administered, and exam score statistics were compiled for evaluation.

Comparison between Cohort K25 and Previous Cohorts

Class	Total Students	Excellent (%)	Good (%)	Average (%)	Below Average (%)	Poor (%)
VL12	67	17 (25%)	33 (49%)	9 (13%)	6 (9%)	2 (4%)
VL6	65	7 (11%)	27 (42%)	30 (46%)	1 (1%)	0 (0%)
VL5	63	4 (6%)	42 (67%)	8 (13%)	9 (14%)	0 (0%)

Table 1. *Exam results of students taught using AI-assisted PBL (Cohort K25)*

Class	Total Students	Excellent (%)	Good (%)	Average (%)	Below Average (%)	Poor (%)
VL09	59	4 (7%)	24 (41%)	21 (36%)	6 (10%)	4 (7%)
VL10	70	3 (4%)	25 (36%)	31 (44%)	10 (14%)	1 (1%)
VL11	69	4 (6%)	36 (52%)	25 (36%)	3 (4%)	1 (1%)

Table 2. *Exam results of students in previous years taught without the new method*

Observations: Statistical analysis of exam and final course grades shows notable improvements: the proportions of Excellent and Good students increased markedly, while Average and Below Average rates declined. In particular, the proportion of Poor-performing students dropped sharply compared to previous academic years.

These findings reaffirm that integrating AI-assisted PBL into teaching yields positive, encouraging outcomes for both student learning methods and faculty instructional practices—fostering independent exploration, autonomous thinking, and creativity to optimize self-directed learning.

4.2. Conclusion

This paper has presented an AI-assisted Problem-Based Learning methodology that enables students to explore independently, sparks passion and interest in Physics, nurtures logical thinking, and maximizes creativity, thereby strengthening self-study and independent research capabilities. Combining PBL with AI in Physics instruction creates a learning environment closely aligned with the specific characteristics of Information Technology. As demonstrated through the illustrative lesson designs, when AI is properly positioned as a simulation and academic dialogue tool under lecturer guidance, students not only master fundamental physical principles but also simultaneously develop engineering problem-solving skills and critical thinking for the digital era.

Through this work, the author hopes that beyond classroom instruction, fellow lecturers will continue exploring this approach to cultivate logical reasoning in students' cognitive processes, enabling them to grasp complex subject matter with greater ease.

References

[1]. Nguyễn Tân Ân (2012). "*Problem-based learning in teaching problem-solving methods of Artificial Intelligence*". *Journal of Science, Hanoi National University of Education*, 57(5), 21–31.
[2]. Lê Thị Thanh Thảo (2005). *"Foundations of modern physics teaching (from theory to practice)*". Postgraduate Training Monograph, Ho Chi Minh City University of Education, 20–22.
[3]. Lương Duyên Bình (2005). *"General Physics Textbook (Vols. 1, 2)*". Vietnam Education Publishing House.
[4]. Hoàng Nghĩa Tý. (2013). *"Data Structures and Algorithms*". Construction Publishing House.
[5]. Đỗ Linh, Lê Văn (2015). "*Effective Learning Methods*". Ho Chi Minh City General Publishing House.