

# Simulación de Movimiento Parabólico para la Intercepción de Misiles

-Juan José Enríquez Córdoba -Samuel Esteban Mena Pupiales

-Santiago Bustos López -Julio Esteban Bolaños Benavides

Universidad Cooperativa de Colombia - Sede Pasto

## Resumen.

Este artículo presenta el desarrollo de una simulación en Python para modelar el movimiento parabólico de un misil y su intercepción mediante un proyectil antiaéreo. Se detallan las ecuaciones que describen la trayectoria de los misiles y los factores que influyen en su comportamiento, tales como la velocidad inicial, el ángulo de lanzamiento y la gravedad. La simulación incluye una interfaz gráfica interactiva que permite ajustar parámetros y observar en tiempo real los resultados. Se realizaron pruebas que demuestran la efectividad del sistema para interceptar el misil en diferentes escenarios.

## Abstract.

This paper presents the development of a Python-based simulation to model the parabolic motion of a missile and its interception by an anti-aircraft projectile. The equations governing the missiles' trajectories are discussed, along with factors such as initial velocity, launch angle, and gravity. The simulation features an interactive graphical interface that allows for real-time parameter adjustments and visualization of the outcomes.

Testing demonstrates the system's effectiveness in intercepting the missile across various scenarios.

## Índice de Términos.

Simulación, Movimiento Parabólico, Intercepción de Misiles, Misil Antiaéreo, Python, Modelado Matemático, Interfaz Gráfica, Trayectoria de Proyectiles, Ecuaciones de Movimiento, Defensa Antiaérea.

## I. Introducción.

En este laboratorio se explorará el concepto de movimiento parabólico a través de la simulación de la trayectoria de un misil. Este fenómeno físico es fundamental para entender las trayectorias de los proyectiles en el contexto militar, donde la precisión y la eficacia en el lanzamiento de misiles son cruciales para el éxito de una operación. Mediante la visualización de la trayectoria parabólica de un misil enemigo y la programación del lanzamiento de un misil antiaéreo para interceptarlo, se busca proporcionar una experiencia práctica que combine teoría y aplicación.

El sistema desarrollado en Python permitirá a los usuarios observar en tiempo real cómo el misil enemigo se desplaza a lo largo de su trayectoria, mientras que el misil antiaéreo se configura para interceptarlo. A través de esta experiencia, se espera que los participantes adquieran un entendimiento más profundo de los principios del movimiento parabólico, así como de las herramientas computacionales necesarias para simularlo.

## II. Objetivos.

Los objetivos de este laboratorio son los siguientes:

1. Comprender las ecuaciones que describen el movimiento parabólico: Estudiar los principios físicos y matemáticos que rigen el movimiento de proyectiles en un entorno gravitacional.
2. Implementar una simulación gráfica de un misil en movimiento: Desarrollar un programa en Python que represente visualmente la trayectoria de un misil enemigo utilizando técnicas de modelamiento matemático y simulación numérica.
3. Diseñar una interfaz interactiva que permita controlar el lanzamiento de un misil antiaéreo: Crear una herramienta que permita a los usuarios modificar parámetros de lanzamiento, facilitando una experiencia de simulación más dinámica.
4. Evaluar si el misil antiaéreo logra impactar al misil enemigo: Establecer un método para determinar la efectividad del misil antiaéreo al impactar en el objetivo,

analizando la coincidencia de las trayectorias.

## III. Desarrollo de las Ecuaciones.

El movimiento parabólico del proyectil hostil se modela a través de dos ecuaciones fundamentales que describen su trayectoria en el espacio tridimensional. Estas ecuaciones permiten determinar la posición del proyectil en función del tiempo, considerando tanto su componente horizontal como vertical.

### Ecuaciones del Movimiento Parabólico.

#### 1. Posición Horizontal:

$$x(t) = v_0 * t * \cos(\theta)$$

En esta ecuación,  $x(t)$  representa la posición horizontal del proyectil en un instante  $t$ . La velocidad inicial  $v_0$  y el ángulo de lanzamiento  $\theta$  son determinantes para calcular el desplazamiento horizontal. Esta relación muestra que el movimiento horizontal es uniforme, es decir, no se ve afectado por la gravedad.

#### 2. Posición Vertical:

$$y(t) = v_0 * t * \sin(\theta) - \frac{1}{2} * g * t^2$$

Aquí,  $y(t)$  es la posición vertical del proyectil en función del tiempo. Esta ecuación tiene en cuenta la influencia de la gravedad  $g$ , que actúa en dirección opuesta al movimiento del proyectil. El término  $v_0 \cdot t \cdot \sin(\theta)$  describe el ascenso del proyectil, mientras que  $-\frac{1}{2}g \cdot t^2$  representa la disminución de altura debido a la gravedad.

### Componentes Clave de las Ecuaciones

- **Velocidad Inicial ( $v_0$ ):** Este parámetro es crucial, ya que determina la rapidez con la que el proyectil es lanzado. Una mayor velocidad inicial resulta en una mayor distancia y altura alcanzada.
- **Ángulo de Lanzamiento ( $\theta$ ):** El ángulo de lanzamiento afecta directamente la forma de la trayectoria del proyectil. En condiciones ideales, un ángulo de  $45^\circ$  maximiza el alcance horizontal.
- **Gravedad ( $g$ ):** La gravedad, aproximadamente  $9.81 \text{ m/s}^2$  es un factor que influye en la duración y la altura del vuelo del proyectil. A medida que el tiempo avanza, el efecto de la gravedad se hace más evidente, causando que el proyectil regrese a la tierra.

### Aplicación de las Ecuaciones en la Simulación.

La implementación de estas ecuaciones permite modelar y simular la trayectoria del proyectil hostil en función del tiempo. Mediante el uso de un entorno de programación, se puede visualizar la trayectoria parabólica, proporcionando una representación gráfica del comportamiento del proyectil a lo largo de su vuelo.

El modelo permite calcular parámetros como la altura máxima alcanzada, la distancia máxima y el tiempo total de vuelo. A partir de estos cálculos, se pueden establecer condiciones para la intercepción de un proyectil interceptador, determinando si es posible interceptar al proyectil hostil en función de su trayectoria.

Con el análisis de estas trayectorias, se busca evaluar la efectividad del sistema de defensa antiaérea, proporcionando información valiosa sobre la dinámica de los proyectiles y las condiciones necesarias para lograr una intercepción exitosa.

## IV. Simulación del Movimiento.

### 1. Implementación de la Función para Graficar la Trayectoria del Misil Enemigo.

El primer paso en la simulación es graficar la trayectoria del misil enemigo. Esto se logra a través del método `graficar_trayectorias()` en la clase `ProyectilInterceptor`, que utiliza la biblioteca `matplotlib` para visualizar la trayectoria de ambos misiles. Aquí se describen las etapas clave de este proceso:

- **Cálculo de la Trayectoria del Proyectil Hostil:**
  - La función `calcular_coordenadas(t)` de la clase `ProyectilHostil` se utiliza para obtener las posiciones  $x$  e  $y$  del misil enemigo a lo largo de su vuelo, para un rango de tiempos calculado mediante `numpy`.
  - Se generan puntos en el tiempo utilizando `np.linspace()` para obtener posiciones en intervalos regulares.
- **Graficar la Trayectoria:**
  - Una vez que se tienen las coordenadas del misil hostil, se

grafican en un sistema de coordenadas.

- Además, se incluye la trayectoria del misil interceptador, que también se calcula utilizando sus respectivas ecuaciones de movimiento.

## 2. Programación del Lanzamiento del Misil Antiaéreo

El segundo componente de la simulación es programar el lanzamiento del misil antiaéreo de tal manera que pueda interceptar al misil enemigo. Este proceso implica varios pasos críticos:

- **Determinación de la Intercepción:**

- La clase `ProyectilHostil` incluye un método llamado `determinar_intercepcion(altura_intercepcion)`, que calcula si es posible interceptar el misil hostil a una altura específica.
- Si la altura de intercepción es mayor que la altura máxima alcanzada por el proyectil hostil se determina que la intercepción no es posible.

- **Cálculo de Velocidad y Ángulo del Misil Antiaéreo:**

- La clase `ProyectilInterceptor` contiene el método `calcular_parametros()`, que utiliza el tiempo de intercepción y las ecuaciones de movimiento parabólico para calcular la

velocidad y el ángulo de lanzamiento necesarios.

- Este método resuelve un sistema de ecuaciones que considera la altura de intercepción y la distancia horizontal desde la posición inicial del interceptador hasta el punto de impacto.

- **Graficar las Trayectorias:**

- Después de calcular los parámetros de lanzamiento, el método `graficar_trayectorias()` de la clase `ProyectilInterceptor` se encarga de visualizar tanto el misil hostil como el interceptador en un mismo gráfico, marcando el punto de intercepción con un símbolo destacado.

## V. Interfaz Gráfica (GUI).

Para facilitar la interacción del usuario con la simulación de interceptación de misiles, se ha implementado una interfaz gráfica utilizando **ipywidgets**. Esta interfaz permite a los estudiantes modificar diversos parámetros del sistema de manera intuitiva y visual. Los parámetros ajustables incluyen:

- **Velocidad Inicial del Misil Enemigo:** Los usuarios pueden establecer la velocidad inicial del misil enemigo mediante un control deslizante que va desde 0 m/s hasta 8000 m/s, con un paso de 100 m/s. Esto permite simular diferentes escenarios de velocidad de ataque.

- **Ángulo de Lanzamiento del Misil Enemigo:** A través de otro control deslizante, los estudiantes pueden ajustar el ángulo de lanzamiento del misil enemigo, que varía entre  $0^\circ$  y  $90^\circ$ , con incrementos de  $1^\circ$ . Este parámetro es crucial para determinar la trayectoria del misil enemigo. Para probar el modelo desarrollado, se propone el siguiente caso de estudio con valores realistas: El misil enemigo es un misil de corto alcance que se lanza de izquierda a derecha hacia un blanco ubicado a 100 km de distancia. Se debe impactar al misil enemigo a una altura de 10 km sobre el suelo. Se deben probar cuatro escenarios:
  1. Velocidad baja y ángulo bajo.
  2. Velocidad baja y ángulo alto.
  3. Velocidad alta y ángulo bajo.
  4. Velocidad alta y ángulo alto.
- **Altura Objetivo de Intercepción del Misil Antiaéreo:** Los usuarios pueden definir la altura a la que se espera que ocurra la intercepción, utilizando un control deslizante que permite seleccionar valores desde 0 m hasta 1,000,000 m, facilitando la simulación de diferentes altitudes de intercepción. En todos estos escenarios, se asegura que el misil enemigo impacte el blanco a 100 km y se valida el funcionamiento del misil antiaéreo para lograr la intercepción en cada caso.
- **Separación Horizontal Inicial:** Este control deslizante permite establecer la distancia inicial entre el misil enemigo y el misil antiaéreo, variando entre 0 m y 16,000 m, lo que es fundamental para simular el contexto de la intercepción. Se deben probar dos escenarios adicionales:
  - a) El misil antiaéreo está ubicado entre el misil enemigo y el blanco.
  - b) El misil antiaéreo está ubicado al lado derecho del blanco.

La interfaz gráfica se actualiza en tiempo real para mostrar las trayectorias del misil enemigo y del misil antiaéreo a medida que los usuarios ajustan los parámetros. Esto proporciona una comprensión visual inmediata de cómo los cambios en los parámetros afectan las trayectorias y la probabilidad de intercepción, permitiendo a los estudiantes experimentar y aprender de manera interactiva antes de realizar el disparo del misil antiaéreo.

## VI. Caso de Estudio.

### Resultados.

Las pruebas realizadas demuestran que el modelo es capaz de interceptar el misil enemigo en todas las configuraciones propuestas. En cada uno de los escenarios, se pudo ajustar los parámetros de velocidad y ángulo, y se logró una intercepción exitosa. La trayectoria del misil antiaéreo se calculó correctamente, permitiendo un impacto en la altura objetivo de 10 km.

Esto valida la funcionalidad del sistema y demuestra que se pueden realizar ajustes en tiempo

real para maximizar la efectividad de la intercepción, incluso en los escenarios más desafiantes.

### **Evaluación de Impacto.**

Para evaluar si el misil antiaéreo logra impactar al misil enemigo, implementamos una condición que verifica si las coordenadas de ambos misiles coinciden en el punto de altura predeterminado. Si las posiciones coinciden dentro de un margen de error aceptable, se considera que el misil antiaéreo ha logrado impactar al enemigo.

Durante el desarrollo del modelo, hemos definido este margen de error con base en las limitaciones físicas del sistema y las tolerancias prácticas de intercepción. Este criterio garantiza que la simulación sea lo más cercana posible a las condiciones reales de funcionamiento.

## **VII. Respuesta a preguntas de análisis**

### **¿Cómo afecta la variación de la velocidad inicial y el ángulo de lanzamiento la trayectoria del misil enemigo?**

La variación de la velocidad inicial y el ángulo de lanzamiento tiene un impacto directo en la forma de la trayectoria del misil enemigo. A mayor velocidad inicial, la trayectoria es más extendida horizontalmente, lo que significa que el misil recorrerá una mayor distancia antes de impactar el blanco. Por otro lado, un ángulo de lanzamiento más elevado incrementa la altura máxima alcanzada por el misil, pero reduce la distancia horizontal recorrida.

En nuestros experimentos, pudimos observar que un ángulo bajo y una velocidad alta resultan en trayectorias más rectas y rápidas, mientras que combinaciones de ángulos altos y bajas velocidades generan trayectorias más arqueadas y lentas. Estos factores son cruciales para ajustar el lanzamiento del misil antiaéreo y lograr la intercepción en el momento y lugar adecuados.

### **¿Qué estrategia utilizamos para lograr la intercepción?**

La estrategia que empleamos se basó en calcular la altura de intercepción, el tiempo necesario para que el misil enemigo alcanzara esa altura y, a partir de ahí, ajustar la velocidad y el ángulo de lanzamiento del misil antiaéreo. Utilizamos ecuaciones que nos permitieron prever la trayectoria del misil enemigo, y en función de esa información, calculamos los parámetros óptimos para el misil interceptador.

Al ajustar constantemente los parámetros de lanzamiento y mediante la implementación de condiciones para verificar la coincidencia de posiciones en el espacio, logramos garantizar que el misil antiaéreo pudiera impactar al enemigo en el punto de intercepción predeterminado, validando así la efectividad del sistema.

## **VIII. Conclusiones.**

A lo largo de esta simulación, logramos obtener una comprensión profunda sobre los factores que influyen en el movimiento parabólico y en la intercepción de misiles. El sistema desarrollado permitió visualizar de manera clara cómo los diferentes parámetros afectan la trayectoria tanto del misil enemigo como del antiaéreo.

Pudimos verificar que el modelo es capaz de ajustarse a distintos escenarios, como variaciones en la velocidad inicial y el ángulo de lanzamiento, logrando intercepciones exitosas. Estos resultados respaldan la validez de las ecuaciones del movimiento parabólico y refuerzan la importancia de ajustar adecuadamente los parámetros de lanzamiento para lograr el impacto.

El éxito de la simulación no solo recae en la precisión matemática de las ecuaciones, sino también en la implementación de una interfaz gráfica que facilita la comprensión y permite experimentar con diferentes configuraciones. Este enfoque práctico nos permitió ver en tiempo real cómo pequeños cambios en los parámetros pueden hacer la diferencia entre una intercepción exitosa y un fallo.

En términos generales, la simulación cumplió con los objetivos propuestos, validando que, en un entorno controlado, se puede diseñar y programar un sistema eficaz para la defensa antiaérea basado en el movimiento parabólico.